POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

Incompressible flows around a square cylinder using adaptive virtual element methods

Relatori
prof. Claudio Canuto
firma dei relatori

. . . .
. . . .

Candidato
Davide Rosso
firma del candidato

. . . .

Anno Accademico 2020-2021

Summary

In the thesis, a Virtual element method (VEM) is applied to a fluid dynamics problem. VEM is a quite recent numerical method that can be applied with general polygonal meshes to solve partial differential equations (PDE) problems. VEM allows the use of "hanging nodes", nodes that divide two aligned edges of a mesh element. In this way in a mesh, elements of the same shape but of different size can coexist close to each other, allowing for a local refinement of the mesh. This is useful for adapting the mesh where the physical problem requires, such as boundary layers in fluid dynamics. The used elements are rectangles with a maximum of one hanging node for each geometrical side to simplify implementation. This constrain is taken into account by the algorithm which refines every mesh element. The accuracy of the adopted VEM discretizations is invariably of second order. The fluid dynamics problem considers an incompressible flow inside a channel containing an obstacle, representing a square cylinder whose centroid is placed on the longitudinal axis. A parabolic velocity profile is assigned at the inflow. The range of Reynolds number, which describes the ratio between convection effects and diffusion ones, is limited so that the flow remains laminar and steady. In the first part of the thesis, a diffusion equation is solved with the VEM, showing error convergence. Moreover adaptive mesh refinement is performed based on an a posteriori error estimator, showing the trends of error and error estimate during the process. The a posteriori error estimator is implemented neglecting one of its part using a recent numerical analysis result. Considering incompressible flows without convection, the discretization of the Stokes problem is implemented. The method is applied to some test cases to confirm error convergence for the gradient of velocity field and for pressure. Adaptive mesh refinement based on an appropriate a posteriori error estimate is implemented, showing behaviour of errors and error estimates during refinement for some cases. In particular the fluid dynamics problem is solved analysing the resulting velocity field and meshes, with hanging nodes, coming from adaptive refinement. Finally, adding the convection terms, the Navier-Stokes problem is solved with the VEM, monitoring error convergence. The model is applied to the fluid dynamics case for different Reynolds numbers in a suitable range. Velocity field and streamlines are shown looking at the flow near the cylinder to understand the situation in relation to the value of Reynolds number. In particular, above a certain Reynolds number eddies can be observed. So results are commented and compared with analogous ones coming from the computational fluid dynamics literature.

Acknowledgements

I'd like to express my deepest gratitude to my supervisor Professor Claudio Canuto (DISMA) for his availability and for his precious advices which helped me in this work. A special thanks goes to Professor Luca Bruno (DAD) that recommended the computational fluid dynamics article related to the fluid dynamics situation considered in the thesis. I'd also like to acknowledge the help of Federico Tesser and Alessandro D'Auria, who taught me how to use the departmental cluster of DISMA.

I'm very grateful to my family for their support during these five years. Finally I'd like to thanks all the people who believed in me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The aim of this thesis is the application of the *Virtual element method (VEM)* to a fluid dynamics problem. VEM is a quite recent numerical method used to solve partial differential equations (PDE), which was introduced in Beirão da Veiga et al. [2013a]. Virtual element method is more general than Finite Element Method in fact it can be used with very general polygonal meshes (Beirão da Veiga et al. [2017]). The term "Virtual" is due to the presence of functions in discrete VEM space which are not known (Beirão da Veiga et al. [2014]), only the degrees of freedom chosen are known for those functions. Furthermore one of the strengths of VEM is the possibility of using meshes with hanging nodes. Hanging nodes are points which separate two edges that have the same direction, namely that are on the same line (Beirão da Veiga et al. [2014]). See for example Figure 1.1.

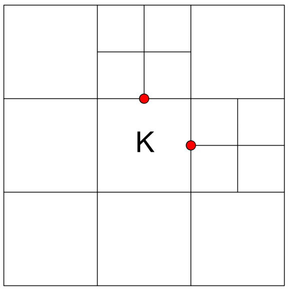


Figure 1.1: The element K has two hanging nodes (the red dots), therefore it is considered an hexagon.

Therefore different size elements can coexist in the mesh. In this way mesh refinement can be done where physics suggests (for example boundary layers in fluid dynamics) or using an error estimate for each element of the grid (Cangiani et al. [2017],Wang et al. [2020]). VEM has already been applied to many physical and engineering problem such as elasticity (Beirão da Veiga et al. [2013b]) and fluid dynamics (Beirão da Veiga et al. [2021a]). Here attention is focused on a meaningful physical situation regarding fluid dynamics.

First of all, to explore VEM method, in Chapter 2 results regarding trend of errors predicted by the theoretical part are discussed. Moreover *adaptive mesh refinement* based on the a posteriori error estimate in Cangiani et al. [2017] is implemented. In Chapter 3, considering incompressible flows without convection, Stokes problem is described showing theoretical results with mathematical tests. Moreover the a posteriori error estimate defined in Wang et al. [2020] is used for adaptive mesh refinement. In particular equations are applied to the fluid dynamics test case described below analyzing the resulting velocity field and how adaptive mesh refinement occurs. Finally convective terms are added in Chapter 4. So the Navier-Stokes equations for an incompressible fluid with different Reynolds numbers are solved using data related to the fluid dynamics situation. Furthermore results are compared with the ones coming from an article in computational fluid dynamics literature (Breuer et al. [2000]).

1.1 Fluid dynamics test case

In order to apply VEM numerical method to a fluid dynamics problem a meaningful test case is taken into account following the work in Breuer et al. [2000]. Here the authors use a Lattice Boltzmann method and a Finite Volume method to realize their simulations.

In particular the situation considered is the flow in a channel with walls and with an inflow and an outflow where proper boundary conditions are imposed. In the middle of the channel there is an obstacle: a square cylinder. The problem is bidimensional, therefore the cylinder section appears as a square. Flows around square cylinders are very used in fluid dynamics literature because when convection is strong vortices and, then, turbulence may appear. However here for simplicity no turbulence situations are considered, therefore flow is **laminar**. In fact, if there is turbulence, Navier-Stokes equations must be solved with a turbulence model which creates a very complex computational problem. Moreover following observations and results of the same article (Breuer et al. [2000]) the Reynold number considered is always below 60. Reynolds number is a fluid dynamics adimensional number which indicates the ratio between inertial effects and diffusion effects. The previous choice is done again to simplify the problem. In fact below $Re = 60$ the flow is **steady**, therefore it is possible to neglect time in equations. In particular the time derivative inside Navier-Stokes equations disappears. Moreover with these limitations and a sufficiently refined mesh no SUPG stabilization is applied to Navier-Stokes equations (for SUPG stabilization for scalar convection problem see Beirão da Veiga et al. [2020]).

Coming to describe in details the physical situation, the square cylinder centroid is placed on the longitudinal axis of the channel. The length of the edge of the cylinder is equal to 1 in adimensional units. The blockage ratio (namely the ratio between the length of cylinder edge and the width of channel) is equal to 1/8 as chosen in Breuer et al. [2000]. More details about the domain used will be added in the geometric description.

The length of the channel is specified for every simulation, anyway in Breuer et al. [2000] this value is 50, in terms of coordinates this means that $x \in [-12, 38]$. For fluid dynamics case simulations here $x \in$ [4,12] in order to save computational time. At $x = 0$ the left vertical edge of cylinder is positioned. In term of y coordinate the domain is restricted to [$-$4,4].

For what concern *boundary conditions*, following Breuer et al. [2000], on the walls as well as on cylinder sides no slip boundary conditions are imposed, so velocity is zero, while a parabolic longitudinal velocity is chosen for inflow boundary. The outflow conditions in Breuer et al. [2000] set velocity components derivative with respect to x equal to zero, here homogeneous Neumann boundary conditions are also considered (See Figure 1.2 for a representation of the fluid dynamics situation).

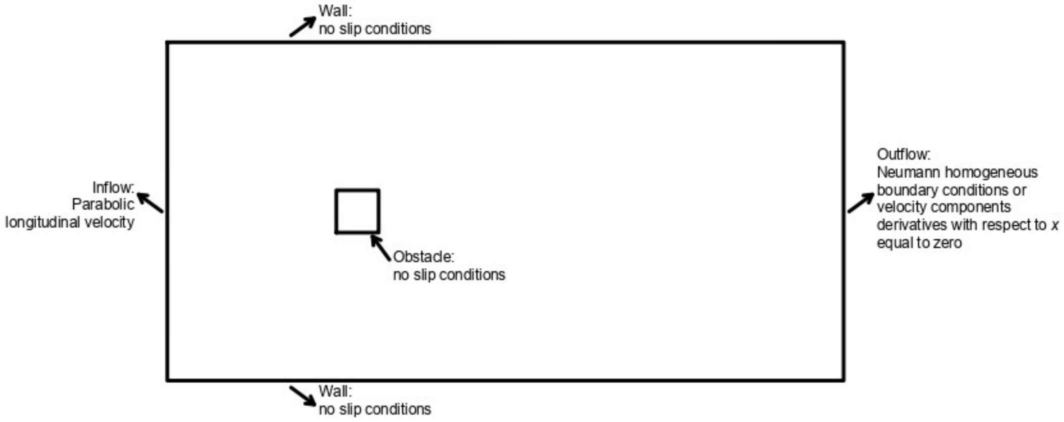


Figure 1.2: Boundary conditions and domain of fluid dynamics test case

Reporting results in Breuer et al. [2000] done with Finite Volume Method (FVM) different situations can happen around the cylinder based on Reynolds number. At $Re \leq 1$ the flow is steady with no separation of laminar boundary layer as diffusion "stabilizes" the flow. Increasing Reynolds number convection becomes more important. In fact at some Reynolds number laminar boundary layer separates at Re_{crit} which is estimated in Breuer et al. [2000] below 5. Above this limit it is possible to see in the wake, in particular at $Re = 30$, a steady recirculation region on the right of cylinder with "two symmetrically placed vortices on each side of the wake" (Breuer et al. [2000]). Length of eddies increases with Re starting from Re_{crit} (Breuer et al. [2000]). Using VEM method for Navier-Stokes equations those cases will be considered and compared to the results in Breuer et al. [2000]. Above a certain Reynolds number $Re \geq 60$ flow becomes unsteady (see Breuer et al. [2000]) but this situation is not taken into account.

Chapter 2

The virtual element method applied to a diffusion problem

The aim of this chapter is the solution of the classical diffusion problem (*Poisson problem*) using virtual element method as described in Beirão da Veiga et al. [2013a]. Particular attention is given to the geometry of the fluid dynamics test case (as discussed in the previous chapter). In addition, an a posterior error analysis is described for *adaptive mesh refinement*.

2.1 The continuous problem

The problem considered (i.e. *Poisson problem*) is

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma = \partial\Omega, \end{cases} \tag{2.1}$$

where $\Omega \subset \mathbb{R}^2$ is a polygonal domain and $f \in L^2(\Omega)$. The solution u is a scalar function and Δ is the Laplacian. In this case the problem chosen to analyze theoretical aspects has Dirichlet homogeneous boundary conditions. Anyway it is possible to apply the method with non-homogeneous boundary Dirichlet conditions or Neumann conditions. Non homogeneous boundary Dirichlet conditions do not modify the formulation of continuous variational problem. In that last case condition $u = 0$ on Γ is replaced by

$$u = g \quad \text{on } \Gamma = \partial\Omega, \tag{2.2}$$

where g is a known function. However such conditions must be considered when solving the discrete system given by the numerical method. This explanation will be done in the implementation part of the chapter.

The problem can be rewritten in a variational formulation:

$$\text{find} \quad u \in V := H_0^1(\Omega) \quad \text{such that} \quad a(u, v) = (f, v) \quad \forall\, v \in V \tag{2.3}$$

where $a(u,v) = (\nabla u, \nabla v)$ and $|v|_1^2 = a(v,v)$, $(\cdot,\cdot)$ is the scalar product in L^2 and $|\cdot|_1$ denotes the seminorm. $a(u,v)$, in this case, is a bilinear form, continuous and coercive, which means:

$$a(u,v) \leq |u|_1|v|_1, \quad a(v,v) \geq |v|_1^2 \quad \forall u,v \in V, \tag{2.4}$$

The seminorm $|\cdot|_1$, is a norm on $H_0^1(\Omega)$ thanks to Poincarè inequality. Under these conditions there exists a unique solution to problem (2.3).

2.2 Virtual element formulation for the diffusion problem

2.2.1 The discrete problem

As done in Beirão da Veiga et al. [2013a], let $\{\mathcal{T}_h\}_h$ be a sequence of decompositions of domain $\Omega \subset \mathbb{R}^2$ into elements K. Elements can be general polygons. However, in all chapters, meshes with rectangular elements will be used allowing the presence of *hanging nodes*. Hanging nodes belong to two consecutive polygon edges which have the same direction (i.e. that are aligned). In this last case, a rectangle is seen as a polygon with more than four edges from the virtual element point of view, even if it remains a rectangle from a geometric point of view. In other words a polygon can admit (as remarked in Beirão da Veiga et al. [2014]) consecutive edges with an angle of 180 degrees, keeping four angles of 90 degrees. In the formulation of the virtual element space, the term "*edge*" always means, if not expressly indicated, the link between two consecutive nodes even if one of them is a hanging node. With *geometrical edge* it is indicated the edge of the rectangle. Figure 1.1 shows an example where the element K has two hanging nodes, therefore it is considered an hexagon by VEM.

For an element K it is possible to define:

$$h_K := \text{diameter of } K, \qquad \mathbf{x}_K := \text{centroid of } K. \tag{2.5}$$

The quantity $h = \max_K h_K$ represents the characteristic length of elements of $\mathcal{T}_h$ decomposition.

Furthermore, for every element K, some geometrical assumptions are done in order to state some theoretical propositions (following Beirão da Veiga et al. [2013a]):

Assumption 1 *There exist a constant $\gamma_1 > 0$ and a constant $\gamma_2 > 0$ such that, for all h*

1. *each element K belonging to $\mathcal{T}_h$ is star-shaped with respect to a ball of radius $\geq \gamma_1 h_K$*
2. *the distance between two vertices of each element K belonging to $\mathcal{T}_h$ is $\geq \gamma_2 h_K$*

These assumptions are satisfied for the mesh and geometry taken in account, therefore in all theorems they are considered as verified.

Moreover the bilinear form $a(\cdot,\cdot)$ and the norm $|\cdot|_1$ can be decomposed:

$$a(u,v) = \sum_{K\in\mathcal{T}_h} a^K(u,v) \quad \forall\, u,v \in V, \quad |v|_1 = \Big(\sum_{K\in\mathcal{T}_h} |v|_{1,K}^2\Big)^{1/2} \quad \forall\, v \in V. \tag{2.6}$$

To formulate the discrete problem some assumptions are done. For each h (so for each decomposition of the domain) such that:

- there exists a space $V_h \subset V$
- there exists a symmetric bilinear form $a_h : V_h \times V_h \to \mathbb{R}$:
$$a_h(u_h, v_h) = \sum_{K \in \mathcal{T}_h} a_h^K(u_h, v_h) \quad \forall\, u_h, v_h \in V_h, \tag{2.7}$$
where $a_h^K(\cdot,\cdot)$ is the bilinear form restricted to the element K.
- there exists an element $f_h \in V_h'$ where V_h' is the dual space of V_h

Now the discrete problem can be stated:

$$\text{find} \quad u_h \in V_h \quad a_h(u_h, v_h) = \langle f_h, v_h \rangle \quad \forall\, v_h \in V_h, \tag{2.8}$$

In order to have the uniqueness of solution u_h some more assumptions have to be done. There exists an integer $k \geq 1$, which is the order of accuracy, for all h and for all K in $\mathcal{T}_h$ such that $\mathbb{P}_k(K) \subset V_h^K$, where V_h^K is the restriction of V_h to K. The order of accuracy used for this problem (and for all the following ones) is $k = 2$. So starting from now k will be always chosen equal to 2.

For the bilinear form $a_h^K(\cdot,\cdot)$ two important properties are assumed:

- *k-Consistency*: $\forall p \in \mathbb{P}_2(K)$ and $\forall v_h \in V_h^K$,
$$a_h^K(p, v_h) = a^K(p, v_h) \tag{2.9}$$
- *Stability*: There exist two positive constants α_* and α^*, independent of h and of K, such that
$$\forall v_h \in V_h^K, \quad \alpha_* a^K(v_h, v_h) \leq a_h^K(v_h, v_h) \leq \alpha^* a^K(v_h, v_h) \tag{2.10}$$

In that situation it is possible to report the following theorem whose proof can be found in Beirão da Veiga et al. [2013a]:

Theorem 1 *If the assumptions done above are verified, the discrete problem (2.8) has a unique solution u_h. Moreover, it exists a constant C depending only on α_* and α^* (2.10) for every approximating function $u_I \in V_h$ of u and for every approximating function u_2 that is piecewise in $\mathbb{P}_2$, such that:*

$$|u - u_h|_1 \leq C(|u - u_I|_1 + |u - u_2|_{h,1} + F_h) \tag{2.11}$$

where $|\cdot|_{h,1}$ is the broken H^1 seminorm:

$$|v|_{h,1} := \Big(\sum_{K \in \mathcal{T}_h} |\nabla v|_{0,K}^2 \Big)^{1/2}, \tag{2.12}$$

while F_h, for every h, is defined as $||f - f_h||_{V_h'}$ or as the smallest constant such that

$$(f, v) - \langle f_h, v \rangle \leq F_h |v|_1 \quad \forall v \in V_h \tag{2.13}$$

In (2.12) $|\cdot|_{0,K}$ is the L^2 norm on element K.

2.2.2 Discretization

Degree of freedom and local virtual element space

For each element (polygon) K it is possible to define a finite-dimensional space for different values of k where k refers to the maximum polynomial degree of polynomials on the edges of the polygon. Here, as already declared, it will be always used $k = 2$. Following the choices done in Beirão da Veiga et al. [2013a] let us define:

$$\mathbb{B}_2(\partial K) := \{v \in C^0(\partial K) : v_{|e} \in \mathbb{P}_2(e) \quad \forall e \subset \partial K\}. \tag{2.14}$$

The total dimensions of this space is $2n$, being n the number of edges of the polygon (which is equal to the number of vertices). The local virtual element finite-dimensional space can be defined:

$$V_h^{K,2} = \{v \in H^1(K) : v_{|\partial K} \in \mathbb{B}_2(\partial K),\, \Delta v_{|K} \in \mathbb{P}_0(K)\}. \tag{2.15}$$

So these functions are polynomials of degree ≤ 2 on ∂K with constant Laplacian in the element. Given a function $w \in \mathbb{B}_2(\partial K)$ and a constant g there is a unique function $u \in H^1(K)$ such that $\Delta u = g$ in K and $u = w$ on ∂K. So the dimension of the local space $V_h^{K,2}$ is $2n + 1$. The degrees of freedom on the space $V_h^{K,2}$, given a function v_h belonging to the space, are:

1. $Dv1$: the values of v_h at the vertices. So, locally, there will be n of these degrees of freedom.

2. $Dv2$: the values of v_h at the midpoint on each edge. So, locally, there will be n of those degrees of freedom.

   Remark 1 *These degrees of freedom, thanks to the choice $k = 2$, are in one-to-one correspondence with the set of edges. This helps the implementation of the method because the index of an edge can identify the correspondent degree of freedom.*

3. $Dv3$: The moment $\frac{1}{|K|}\int_K v_h(\mathbf{x})d\mathbf{x}$ where $|K|$ is the area of element K. Globally every one of these degrees are in one-to-one correspondence with the set of elements.

In Beirão da Veiga et al. [2013a] the following fundamental proposition is stated and demonstrated.

Proposition 1 *Let K be a simple polygon with n edges and let the space $V_h^{K,2}$ be defined as in (2.15). The degrees of freedom $Dv1$ plus $Dv2$ plus $Dv3$ are unisolvent for $V_h^{K,2}$. So they determine uniquely an element of $V_h^{K,2}$.*

Remark 2 *The degrees of freedom $Dv1$ and $Dv2$ are a consequence of $v_{h|\partial K} \in \mathbb{B}_2(\partial K)$. Indeed, they uniquely determine an element of $\mathbb{B}_2(\partial K)$. The degree of freedom $Dv3$ is equivalent to indicate $P_0^K v_h$ where P_0^K is the L^2-projection in K on the constants space (see Beirão da Veiga et al. [2013a]).*

It is possible to define, for each element K, the canonical basis $\{\varphi_i\}_{i=1\dots N^k}$ (indicating with N^k the number of local degrees of freedom), which satisfies:

$$\mathrm{dof}_i(\varphi_j) = \delta_{ij}, \tag{2.16}$$

where $\mathrm{dof}_i(\cdot)$ indicates the $i - th$ degree of freedom and δ the Kronecker delta.

Remark 3 *It is fundamental to underline that the functions φ_i, defined above, are not known. Indeed the term "virtual" of method derives from the fact that some of the functions in the space are not known (see Sutton [2017] and Beirão da Veiga et al. [2014]).*

Global virtual element space V_h

For every decomposition $\mathcal{T}_h$ of the domain Ω, the global virtual element space on the whole Ω is defined as:

$$V_h = \{v \in V : v_{|\partial K} \in \mathbb{B}_2(\partial K) \quad \text{and} \quad \Delta v_{|K} \in \mathbb{P}_0(K) \quad \forall K \in \mathcal{T}_h\}. \tag{2.17}$$

This space has dimension equal to

$$\dim V_h = N_{dofs} = N_V + N_E + N_P, \tag{2.18}$$

where N_V, N_E and N_P (respectively) are the total number of *internal* (as Dirichlet boundary conditions are here considered) vertices, edges and elements of $\mathcal{T}_h$. The global degrees of freedom chosen (that can be obtained from the local ones) are (respectively):

1. $\mathcal{V}$: The values of v_h at the internal vertices
2. $\mathcal{E}$: The values of v_h at the midpoint of each internal edge. (So they are in one-to-one correspondence with the set of edges).
3. $\mathcal{P}$: The moments $\frac{1}{|K|}\int_K v_h(\mathbf{x})d\mathbf{x} \quad \forall$ element K.

The fact that $V_h \in V$ implies that the value of v_h on the boundary is given by Dirichlet conditions.

It is possible to define a global canonical basis on global space with the same definition as in (2.16) $\{\varphi_i\}_{i=1\dots N_{dofs}}$ using those degrees of freedom. These degrees of freedom are unisolvent for the global space V_h (Beirão da Veiga et al. [2013a]).

Theoretical results

Thanks to the fact that geometrical assumptions, written above, are satisfied in this case, it is possible to report the following propositions stated in Beirão da Veiga et al. [2013a] following Brenner and Scott [2008].

Projection error

Proposition 2 *There exists a constant C (which depends only on the order of the VEM space ($k = 2$) and the constant γ_1 in Assumption 1), such that $\forall s$ with $1 \le s \le 3$ and $\forall w \in H^s(K)$ there exists $w_\pi \in \mathbb{P}_2(K)$ s.t.:*

$$||w - w_\pi||_{0,K} + h_K|w - w_\pi|_{1,K} \le Ch_K^s|w|_{s,K} \tag{2.19}$$

where $|w|_{s,K}$ is the seminorm of order s restricted to the element K.

Interpolation error

Proposition 3 *There exists a constant C (which depends only on the order of VEM space ($k = 2$) and on the constant γ_1 in Assumption 1), such that $\forall s$ with $1 \leq s \leq 3$ and $\forall w \in H^s(K)$ there exists $w_I \in V_h^{K,2}$ s.t.:*

$$||w - w_I||_{0,K} + h_K|w - w_I|_{1,K} \leq Ch_K^s|w|_{s,K} \tag{2.20}$$

Building the bilinear form a_h

To build the discrete bilinear form $a_h(\cdot, \cdot) : V_h \times V_h \to \mathbb{R}$, following Beirão da Veiga et al. [2013a], it is necessary to define the so-called *Nabla operator*.

Nabla operator Given an element K of the decomposition $\mathcal{T}_h$, the operator $\Pi_2^{\nabla,K}$: $V_h^{K,2} \to \mathbb{P}_2(K) \subset V_h^{K,2}$ is defined, $\forall v \in V^{K,2}$, as the solution of the system:

$$\begin{cases} a^K(\Pi_2^{\nabla,K} v, q) = a^K(v, q) \quad \forall q \in \mathbb{P}_2(K), \\ P_0^K(\Pi_2^{\nabla,K} v) = P_0^K(v), \end{cases} \tag{2.21}$$

where P_0^K is the $L^2(K)$ projection on constants space. As in Beirão da Veiga et al. [2014] it is defined as:

$$P_0^K(v) = \frac{1}{|K|}\int_K v(\mathbf{x})d\mathbf{x} \tag{2.22}$$

The first equation of (2.21) imposes a condition on the gradient $\Pi_2^{\nabla,K} v$ while the second equation is necessary to control the constant part. The definition (2.21) implies:

$$\Pi_2^{\nabla,K} q = q \quad \forall q \in \mathbb{P}_2(K). \tag{2.23}$$

Note that if $q \in \mathbb{P}_2(K)$, then $a^K(q, v)$ is computable through an integration by part:

$$a^K(q, v) = \int_K \nabla q \cdot \nabla v \, dK = -\int_K \Delta q v \, dK + \int_{\partial K} \nabla q \cdot \mathbf{n} \, v \, ds; \tag{2.24}$$

since $\Delta q \in \mathbb{P}_0(K)$, namely it is constant, the first integral can be computed using moment of order zero of v. On edge e, $\nabla q \cdot \mathbf{n}(e) \in \mathbb{P}_1(e)$ and $v \in \mathbb{P}_2(e)$ for every edge $e \subset \partial K$; thus the second integral can be computed using $Dv1$ and $Dv2$.

The bilinear form $a_h^K(u, v) = a^K(\Pi_2^{\nabla,K} u, \Pi_2^{\nabla,K} v)$ satisfies the *k-consistency* property (2.9) but in general not the *stability* one (2.10). Therefore another term $S^K(u, v)$ is added. $S^K(u, v)$ is chosen as any symmetric positive definite bilinear form which verifies:

$$c_L a^K(v, v) \leq S^K(v, v) \leq c_U a^K(v, v) \quad \forall v \in V_h^{K,2} \quad \text{with} \quad \Pi_2^{\nabla,K} v = 0, \tag{2.25}$$

for some positive constants c_L and c_U independent of K. So the discrete bilinear form a_h is defined in the following way:

$$a_h^K(u, v) = a^K(\Pi_2^{\nabla,K} u, \Pi_2^{\nabla,K} v) + S^K(u - \Pi_2^{\nabla,K} u, v - \Pi_2^{\nabla,K} v) \quad \forall u, v \in V_h^{K,2}. \tag{2.26}$$

This definition imitates the identity:

$$a^K(u,v) = a^K(\Pi_2^{\nabla,K} u, \Pi_2^{\nabla,K} v) + a^K(u - \Pi_2^{\nabla,K} u, v - \Pi_2^{\nabla,K} v) \quad \forall u,v \in V_h^{K,2}, \tag{2.27}$$

in which the case $u = v$ corresponds to Pythagoras theorem. The following theorem stated and demonstrated in Beirão da Veiga et al. [2013a] is valid.

Theorem 2 *The form a_h defined in (2.26) satisfies the consistency and stability properties.*

Choice of S^K

Following what is done in Beirão da Veiga et al. [2013a] and Beirão da Veiga et al. [2014], S^K is chosen as:

$$S^K(\varphi_i - \Pi_2^{\nabla,K}\varphi_i, \varphi_j - \Pi_2^{\nabla,K}\varphi_j) = \sum_{r=1}^{N^K} \mathrm{dof}_r(\varphi_i - \Pi_2^{\nabla,K}\varphi_i)\mathrm{dof}_r(\varphi_j - \Pi_2^{\nabla,K}\varphi_j) \tag{2.28}$$

where $\{\varphi_i\}_{i=1\dots N^k}$ are the local canonical basis functions. This definition must satisfy (2.25) and to do so $S^K(\cdot,\cdot)$ must scale like $a^K(\cdot,\cdot)$. Under a geometrical assumption (item 2 of Assumption 1) which is satisfied, it is possible to notice that $a^K(\varphi_i,\varphi_i) \simeq 1 \quad \forall i$. Therefore condition (2.25) is valid with this choice (2.28), which is called dofi-dofi stabilization (Beirão da Veiga et al. [2021a]).

Right-hand side and term f_h

Following Beirão da Veiga et al. [2013a], the term f_h is defined as:

$$f_h = P_0^K f = \frac{1}{|K|}\int_K f\,dK \quad \text{on each} \quad K \in \mathcal{T}_h \tag{2.29}$$

Therefore, the right-hand side will be

$$\langle f_h, v_h\rangle = \sum_{K\in\mathcal{T}_h}\int_K f_h\, v_h\, dK = \sum_{K\in\mathcal{T}_h}\int_K P_0^K\, f v_h\, dK = \sum_{K\in\mathcal{T}_h}\int_K f\, P_0^K\, v_h\, dK. \tag{2.30}$$

Now, it is possible to write:

$$\langle f_h, v_h\rangle = \sum_{K\in\mathcal{T}_h} P_0^K v_h \int_K f\, dK = \sum_{K\in\mathcal{T}_h} \frac{1}{|K|}\int_K v_h \int_K f\, dK \tag{2.31}$$

It can be noticed that $P_0^K v_h$ corresponds to the momentum degree of freedom.

It is possible to show that F_h, defined in Equation (2.13), satisfies:

$$F_h \leq Ch^2 \left(\sum_{K\in\mathcal{T}_h} |f|_{1,K}^2\right)^{1/2}. \tag{2.32}$$

Combining (2.11) with (2.32) it is possible to write:

Remark 4 *The H^1-norm error, so the gradient error, satisfies:*

$$|u - u_h|_1 = \|\nabla u - \nabla u_h\|_0 \simeq O(h^2) \tag{2.33}$$

if $u \in H^3(\Omega)$. $||\cdot||_0$ is the L^2 norm.

2.3 Geometry of the domain and mesh

In this section the domain of interest, coming from fluid dynamics as explained in Chapter 1, as well as the meshes used in the thesis, are described.

Domain The domain chosen aims to describe a real situation which can be meaningful in fluid dynamics problem. The situation is built looking at the fluid dynamics literature, in particular at the work done in Breuer et al. [2000]. The region occupied by the fluid is a bidimensional rectangle, from which a square "hole" is removed (see Figure 2.1). The rectangle is given by $[-A, L] \times [-4,4]$, where A and L will be specified in numerical tests and experiments. The hole corresponds to the section of a square cylinder which can be considered as an obstacle for the flow. Its left bottom corner is positioned in $(0, -1/2)$ and the length of the edge is equal to 1. In every piece of boundary of the domain, including the ones given by the obstacle, proper conditions must be set and taken into account when solving the problem. From a fluid dynamics point of view the horizontal external boundaries represent the walls of a channel while the other edges of external boundary are the inflow and outflow boundaries. However most of numerical tests use "mathematical" boundaries condition which need not correspond to physical situations.

Mesh The meshes considered are always composed by elements which are geometrically squares or rectangles. However in a mesh there can be elements of different areas. In fact the element refinement process splits the "father" element in 4 "children" rectangles connecting the centroid with midpoints of every edge. In order to properly refine, as already underlined, hanging nodes are permitted thanks to the properties of virtual element approach. The presence of hanging nodes does not change the shape of an element, they always remain rectangles. However, from the point of view of the virtual element method, a squared element of the mesh with an hanging node appears, for example, as a five sides polygon (so a pentagon). In Figure 2.1 a uniform mesh is shown, while in Figure 2.2 a mesh with elements of different shapes and hanging nodes is shown in the domain. The implementation of the mesh is done in the script *mesh_generation_test_case*.

Assumption 2 *The presence of hanging nodes is admitted. However every element can have at most one hanging node per edge. Therefore an element cannot have more than 8 vertices. This constrain creates a mesh where "small" element cannot be near a "big" element.*

This assumption is a limitation that was chosen in order to simplify the implementation of refinement algorithm and meshes which can occur in refinement. This assumption is also done in Popinet [2003]. The mesh is saved in a *structure* of *Matlab*. Its various fields describe:

- *vertices*: their coordinates in the 2D domain
- *edges* : the starting and the final vertex which define an edge
- *elements*. Every element is defined with the indices of its vertices in counterclockwise order. Moreover different fields are associated to every element:

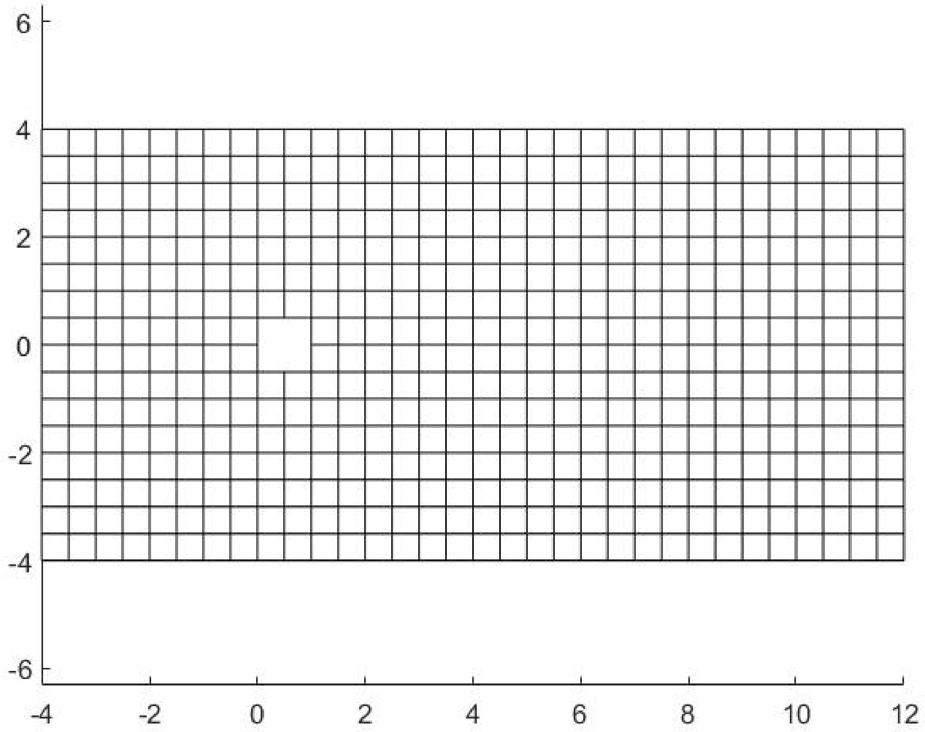


Figure 2.1: Example of square mesh with $A = 4$ and $L = 12$. This is the coarsest mesh considered in the thesis.

1. *.vertices* collects the indices of the vertices of polygon. To distinguish geometrical nodes from hanging nodes, a matrix with two rows of four entries is used. A positive integer in the second row is the index of the hanging node that is found after the geometrical node reported in the same matrix column. If the hanging node is not present, a negative number (-1) is used instead of the index.

   In Table 2.1 it is shown the vertices matrix considering element 4 of the example-mesh in Figure 2.3.

5	6	10	9
-1	-1	17	-1

   Table 2.1: Vertices field of element 4 of example-mesh of Figure 2.3

2. *.edges* collects the indices of edges in counterclockwise order. Two edges can have the same direction due to the presence of hanging nodes. An edge, in fact, links two following vertices and does not correspond necessarily to the geometrical edge of a rectangle. The same method adopted to save vertices

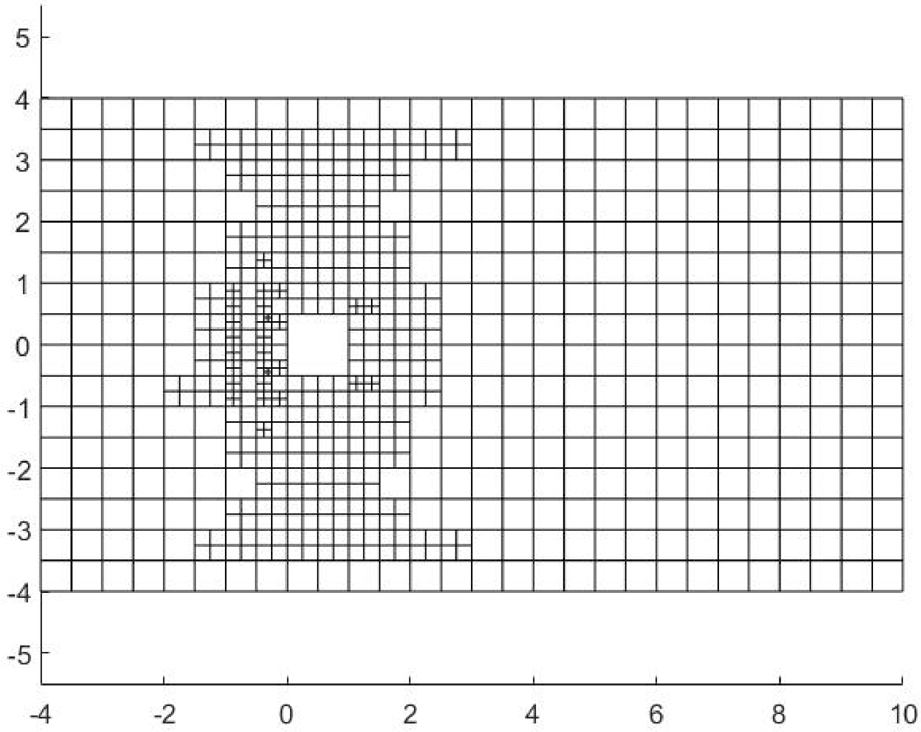


Figure 2.2: Example of a square mesh with $A = 4$ and $L = 12$. Some elements have been refined: hanging nodes are present in the zone around the "hole" (the cylinder)

is used to save edges, in fact for every geometrical boundary there could be a maximum of two edges (thanks to Assumption 2).

Table 2.2 shows the edges matrix considering element 4 of the example-mesh in Figure 2.3.

4	17	25	14
-1	-1	7	-1

Table 2.2: Edges field of element 4 of example-mesh of Figure 2.3 (indices of edges are not shown in the figure)

3. *.neigh* collects the indices of neighboring elements which have at least one edge in common with the current element. Again they are saved following an counterclockwise order. Thanks to Assumption 2 for every "geometrical" edge of the rectangle there can be two elements adjacent to the current element. Therefore the neighboring elements are saved, again, in the same way of the vertices and so in a 2×4 matrix where every column contains the neighbouring elements

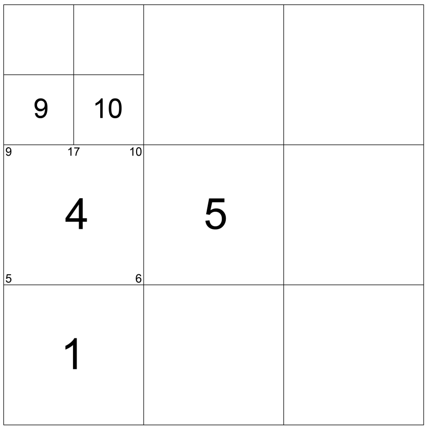


Figure 2.3: Example of part of a mesh showing index of an element (4), the indices of its vertices and of its neighbouring elements

related to one of the four geometrical sides. If a neighbouring element does not exist because the current one confines with the external boundary or the obstacle a "0" is saved.

In Table 2.3 it is shown the neighbouring elements matrix considering element 4 of the example-mesh in Figure 2.3.

1	5	10	0
0	0	9	0

Table 2.3: Neigh field of element 4 of example-mesh of Figure 2.3

4. *.level* indicates "how much" an element has been refined starting from 0 (not refined even once). When an element is generated splitting a previous one (every element generates 4 new elements), after a refinement of the mesh, its level is given by the level of the "father" element plus one. Assumption 2 implies the following observation.

   Remark 5 *The level difference between two adjacent elements is always* ≤ 1.

   For example element 4 in Figure 2.3 has a level equal to the level of element 10 minus 1.

- *.boundary* collects the indices of vertices and edges belonging to external boundary. There is a different field for every boundary edge of external rectangle. Indices are saved following an counterclockwise order.

- *.obstacle* collects the indices of vertices and edges of the square section of the cylinder (the obstacle) in counterclockwise order.

2.4 Set-up of the linear system and details about implementation

In this section, the implementation of the virtual element discretization of the diffusion problem is described. It is important to underline that our implementation follows the guidelines contained in Beirão da Veiga et al. [2014] and Sutton [2017] and the code to obtain VEM solution of diffusion problem is written adapting the one in Sutton [2017]. The numerical method results in solving a linear system of equations as for the finite-element method (see for example Quarteroni [2016]). Considering the global canonical Lagrangian basis $\{\varphi_j\}_{j=1,...,N_{dofs}}$ of space V_h, it is possible to restate the discrete problem as follows:

$$\text{find} \quad u_h \in V_h : a_h(u_h, \varphi_j) = (f_h, \varphi_j) \quad \forall j = 1, ..., N_{dofs} \tag{2.34}$$

Moreover $u_h \in V_h$, hence,

$$u_h = \sum_{i=1}^{N_{dofs}} u_i \varphi_i \tag{2.35}$$

The previous equation is not valid in case of non-homogeneous Dirichlet boundary conditions. In fact a term $\sum_{j \in N_{\partial\Omega}} u_j \varphi_j$ must be added, where $N_{\partial\Omega}$ is the set of "degrees of freedom" on the Dirichlet boundary. φ_j are functions related to vertices and midpoints of edges of Dirichlet boundaries, following the same definition as canonical basis functions. u_j are the values of function g (see (2.2)) evaluated in vertices and midpoints of each edge on the boundary.

So the problem (2.34) can be rewritten as algebraic system:

$$\mathbf{A}\mathbf{u} = \mathbf{f} \tag{2.36}$$

where

$$\mathbf{A}_{ji} = a_h(\varphi_i, \varphi_j) \qquad \mathbf{f}_j = (f_h, \varphi_j) \tag{2.37}$$

$\mathbf{u}$ is the array containing components of solution u respect to global canonical basis and $\mathbf{A}$ is the global stiffness matrix. The matrix and vector are obtained assembling the local ones.

Local stiffness matrix

The local stiffness matrix will be given by $\mathbf{A}^K_{ji} = a^K_h(\varphi_i, \varphi_j)$. Following the construction done in Beirão da Veiga et al. [2014] let be on the element K the set of scaled monomials (also done in Sutton [2017]):

$$\mathcal{M}_k(K) = \left\{ \left(\frac{\mathbf{x} - \mathbf{x}_K}{h_K} \right)^{\alpha} : 0 \leq |\alpha| \leq k \right\} \tag{2.38}$$

where α is the multiindex: $\alpha = (\alpha_1, \alpha_2)$ with $|\alpha| = \alpha_1 + \alpha_2$, so if $\mathbf{x} = (x_1, x_2)$ then $\mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2}$. In the case of $k = 2$ the elements of $\mathcal{M}_2(K)$ will be:

$$\left\{1, \frac{x - x_K}{h_K}, \frac{y - y_K}{h_K}, \left(\frac{x - x_K}{h_K}\right)^2, \left(\frac{y - y_K}{h_K}\right)^2, \left(\frac{x - x_K}{h_K}\right)\left(\frac{y - y_K}{h_K}\right)\right\} \tag{2.39}$$

This is chosen as a polynomial basis of the space $\mathbb{P}_2(K)$ where x and y are the cartesian coordinates. When using them for matrix assembling basis elements are numerated following the order they are written above.

Considering the elements of local canonical basis $\{\varphi_i\}_{i=1 \ldots N^K}$, since Nabla operator $\Pi_2^{\nabla,K} \varphi_i \in \mathbb{P}_2(K)$ it possible to write it respect to the polynomial basis $\mathcal{M}_2(K)$:

$$\Pi_2^{\nabla,K} \varphi_i = \sum_{\beta=1}^{N^K} s_i^\beta m_\beta \tag{2.40}$$

where N^K is the number of local degrees of freedom of element K and $i = 1, ..., N^K$. From the definition of Nabla operator (2.21) it is possible to write:

$$\sum_{\beta=1}^{N^K} s_i^\beta a^K(m_\alpha, m_\beta) = a^K(m_\alpha, \varphi_i), \quad \text{for} \quad \alpha = 2, ..., 6 \tag{2.41}$$

which is valid for $i = 1, ..., N^K$.

The previous equation is not considered for $\alpha = 1$ (namely when the monomial is $m_1 = 1$) because it would be equivalent to the identity $0 = 0$. Therefore one more equation is given by the part of definition of Nabla operator related to P_0^K. In fact it is true that:

$$\sum_{\beta=1}^{N^K} s_i^\beta P_0^K m_\beta = P_0^K \varphi_i \tag{2.42}$$

In this way the system of equations (coming from (2.41) and (2.42)) can be rewritten in matrix form:

$$\mathbf{G}\mathbf{s}_i = \mathbf{b}^i \tag{2.43}$$

where $\mathbf{s}_i^\beta = s_i^\beta$ and $\mathbf{G} \in \mathbb{R}^{6\times 6}$ is the matrix such that:

$$\begin{cases} \mathbf{G}_{\alpha\beta} = a^K(m_\alpha, m_\beta) & \text{for} \quad i \neq 1 \\ \mathbf{G}_{1\beta} = P_0^K m_\beta & \text{for} \quad i = 1. \end{cases} \tag{2.44}$$

Given $1 \leq i \leq N^K$, $\mathbf{b}^i$ is a vector such that:

$$\begin{cases} \mathbf{b}_\beta^i = a^K(m_\beta, \varphi_i) & \text{for} \quad \beta \neq 1 \\ \mathbf{b}_1^i = P_0^K \varphi_i & \text{for} \end{cases} \tag{2.45}$$

In order to write equations in a compact form with all i let be:

$$\mathbf{B} := \left[\mathbf{b}^1, ..., \mathbf{b}^{N^K}\right] \tag{2.46}$$

Therefore

$$\begin{cases} \mathbf{B}_{\alpha\beta} = a^K(m_\alpha, \varphi_\beta) & \text{for} \quad \alpha \neq 1 \\ \mathbf{B}_{1\beta} = P_0^K \varphi_\beta \end{cases} \tag{2.47}$$

So $\mathbf{B} \in \mathbb{R}^{6\times N^K}$.

Now it is possible to build the matrix representation $\mathbf{\Pi}_*^\nabla \in \mathbb{R}^{N^K\times 6}$ of Nabla operator $\Pi_2^{\nabla,K}$ acting from $V_h^{K,2}$ to $\mathbb{P}_2(K)$ (described using polynomial basis $\mathcal{M}_2(K)$). Therefore $(\mathbf{\Pi}_*^\nabla)_{\beta i} = s_i^\beta$ and it can be obtained from:

$$\mathbf{\Pi}_*^\nabla = \mathbf{G}^{-1}\mathbf{B} \tag{2.48}$$

Since $\mathbb{P}_2(K) \subset V_h^{K,2}$, $\Pi_2^{\nabla,K}$ can be seen as a operator acting from $V_h^{K,2}$ to $V_h^{K,2}$ (described using the usual local canonical basis). Given an element of local canonical basis φ_i:

$$\Pi_2^{\nabla,K}\varphi_i = \sum_{j=1}^{N^K} t_i^j \varphi_j, \quad \text{for} \quad i = 1, ..., N^K \tag{2.49}$$

where:

$$t_i^j = \mathrm{dof}_j(\Pi_2^{\nabla,K}\varphi_i) \tag{2.50}$$

Let now be the matrix $\mathbf{D} \in \mathbb{R}^{N^K\times 6}$ defined as:

$$\mathbf{D}_{i\alpha} = \mathrm{dof}_i(m_\alpha) \quad \text{for} \quad i = 1, ..., N^K \tag{2.51}$$

where dof_i is $i - th$ local degree of freedom while m_α is an element of polynomial basis as defined before.

Now it is possible to write matrix representation $\mathbf{\Pi}^{\nabla,K}$ of operator $\Pi^{\nabla,K}$ (acting from $V_h^{K,2}$ to $V_h^{K,2}$) as:

$$\mathbf{\Pi}^{\nabla,K} = \mathbf{D}\mathbf{\Pi}_*{}^{\nabla,K} = \mathbf{D}\mathbf{G}^{-1}\mathbf{B} \tag{2.52}$$

The detailed derivation of the previous formula is shown in Beirão da Veiga et al. [2014].

The matrix $\mathbf{G}$ can be also calculated using $\mathbf{B}$ and $\mathbf{D}$:

$$\mathbf{G} = \mathbf{B}\mathbf{D} \tag{2.53}$$

The proof of this equivalence can be found again in Beirão da Veiga et al. [2014].

As observed in Beirão da Veiga et al. [2014] matrices $\mathbf{B}$ and $\mathbf{D}$ and matrix $\mathbf{G}$ can be computed separately, then using property (2.53) the correctness of the results can be verified. Moreover $\mathbf{G}$ does not depend on the elements geometries or degrees of freedom but only on the monomials basis and so by method's order ($k = 2$). In case of simple rectangles and $k = 2$ matrices $\mathbf{B}$, $\mathbf{D}$ and $\mathbf{G}$ implemented could be verified looking at calculation reported in Beirão da Veiga et al. [2014].

Remembering the definition of bilinear form $a_h^K(\cdot,\cdot)$ (2.26) and the choice of stabilizing form S^K (2.28) it is possible to write the local stiffness matrix $\mathbf{A}^K$ as:

$$(\mathbf{A}^K)_{ij} = a^K(\Pi_2^{\nabla,K}\varphi_i, \Pi_2^{\nabla,K}\varphi_j) + \sum_{\alpha=1}^{N^K} \mathrm{dof}_\alpha((\mathbf{I} - \Pi_2^{\nabla,K})\varphi_i)\mathrm{dof}_\alpha((\mathbf{I} - \Pi_2^{\nabla,K})\varphi_j) \tag{2.54}$$

Therefore, as shown in Beirão da Veiga et al. [2014], it can be derived that the following matrix equivalence is true:

$$\mathbf{A}^K = (\mathbf{\Pi}_*^{\nabla,\mathbf{K}})^T \tilde{\mathbf{G}}(\mathbf{\Pi}_*^{\nabla}) + (\mathbf{I} - \mathbf{\Pi}^{\nabla,K})^T(\mathbf{I} - \mathbf{\Pi}^{\nabla,K}) \tag{2.55}$$

where $\tilde{\mathbf{G}}_{ij} = \mathbf{G}_{ij}$ $\forall i = 2, ..., 6$ and $\forall j = 1, ..., 6$; therefore the only difference between the two matrices is the first row. Moreover $\tilde{\mathbf{G}}_{1j} = 0$ $\forall j = 1, ..., 6$; so the first row is all zero.

Remark 6 *Every time an integral needs to be computed on the edges of polygons Simpson quadrature rule is used (see Monegato [2008]). Given a scalar function $v(x)$ defined on a interval $[a, b]$:*

$$\int_a^b v(x)\,dx = \frac{h}{3}\left(f(a) + 4f(a+h) + f(b)\right) + R(f) \tag{2.56}$$

where $h = \frac{b-a}{2}$:

$$R(f) = -\frac{b-a}{180} h^4 f^{(4)}(\eta), \quad a < \eta < b, \quad f \in C^4[a, b] \tag{2.57}$$

However all functions inside integrals result in polynomials of degree < 4 so the formula is computationally exact.

This formula is particularly suitable for the degrees of freedom used. In fact it is possible to know value of a function on initial, final and medium point of an edge. Indeed, those points are the ones used in Simpson formula.

Right-hand side

In order to compute $\langle f_h, v_h \rangle$ as described in (2.31) a Simpson quadrature rule formula generalized to two dimensions is used to approximate integral of f on quadrangular element K:

$$\int_K f(\mathbf{x})d\mathbf{x} \simeq \frac{|K|}{36}\left(\sum_{i=1}^{4} f(\mathbf{x}_i^V) + 4\sum_{j=1}^{4} f(\mathbf{x}_j^M) + 16 f(\mathbf{x}^C)\right) \tag{2.58}$$

where $\mathbf{x}^V$ are coordinates of vertices of the rectangle, $\mathbf{x}^M$ are coordinates of midpoint of each edge of the rectangle and $\mathbf{x}^C$ are the coordinates of centroid. The Simpson quadrature rule is of fourth order (Monegato [2008]), therefore if f is a polynomial of degree ≤ 3 the computation of integral is exact.

For what concerns $\int_K \varphi_i$ it is non-zero only for canonical basis function corresponding to the degree of freedom associated to the momentum of zero-order in K. Therefore if i indicates that degree of freedom:

$$\frac{1}{|K|}\int_K \varphi_i = 1 \tag{2.59}$$

from the definition of canonical basis.

So the local vector $\mathbf{f}^K$ will be all zero except for the entry related to zero-order momentum degree of freedom. In this last case the value will be given by (2.58).

Non-homogeneous Dirichlet boundary conditions

To deal with non-homogeneous Dirichlet boundary conditions the VEM stiffness matrix and the right-hand side are computed considering also "false" degrees of freedom corresponding to values of a function $v_h \in V_h$ on vertices and midpoint of each edge belonging on the boundaries of the domain. Then, when solving the final linear system, these degrees of freedom are deactivated and an additional term, due to non-homogeneous Dirichlet boundary condition, is added. Indicating with *internal_dofs* the true "degrees of freedom", which are not on the boundary, and with *boundaries* the "false" degrees of freedom related to the boundary, the final matrix equation (as explained in Sutton [2017]):

$$\begin{aligned}\mathbf{A}(internal\_dofs, internal\_dofs)\mathbf{u_h}(internal\_dofs) = \\ \mathbf{f} - \mathbf{A}(internal\_dofs, boundaries)\mathbf{u_h}(boundaries)\end{aligned} \tag{2.60}$$

H_0^1 norm error (gradient error)

(2.33) explains how H_0^1 norm error (gradient error) depends on h. However u_h is not known inside the domain Ω and so the error is defined as (see Beirão da Veiga et al. [2017] for the analogous case):

$$\left(\sum_{K \in \mathcal{T}_h} ||\nabla u - \mathbf{\Pi}_1^{0,K} \nabla u_h||_0^2 \right)^{1/2} \tag{2.61}$$

where $\mathbf{\Pi}_1^{0,K}$ is the $L^2(K)$-projector for vectorial functions on $[\mathbb{P}_1(K)]^2$. Given a vector function $\mathbf{v}$ it is defined as (see Beirão da Veiga et al. [2017]):

$$\int_K (\mathbf{v} - \mathbf{\Pi}_1^{0,K}\mathbf{v}) \cdot \mathbf{m}_1 d\mathbf{x} = \mathbf{0} \quad \forall \mathbf{v} \in [L^2(K)]^2 \quad \text{and} \quad \forall \mathbf{m}_1 \in [\mathbb{P}_1(K)]^2 \tag{2.62}$$

In order to compute (2.61) the following basis for the vector space $[\mathbb{P}_1(K)]^2$:

$$\mathcal{M}_1^2(K) = \{ \begin{bmatrix} m_1^i \\ 0 \end{bmatrix} \quad \text{for} \quad i = 1, ..., 3 \quad ; \begin{bmatrix} 0 \\ m_1^j \end{bmatrix} \quad \text{for} \quad j = 1, ..., 3 \} \tag{2.63}$$

where $m_1^i \in \mathcal{M}_1(K)$ and $m_1^j \in \mathcal{M}_1(K)$. So this space has 6 elements.

Given the gradient of an element $u_h \in V_h$, its projector can be written in the following way:

$$\mathbf{\Pi}_1^{0,K} \nabla u_h = \sum_{j=1}^{6} s_j \mathbf{m}_j \quad \text{where} \quad \mathbf{m}_j \in \mathcal{M}_1^2(K) \tag{2.64}$$

Therefore indicating with $\mathbf{m}_j$ the elements of basis $\mathcal{M}_1^2(K)$ it is possible to rewrite (2.62) with $\mathbf{v} = \nabla u_h$:

$$\sum_{j=1}^{6} s_j \int_K \mathbf{m}_z \cdot \mathbf{m}_j \, d\mathbf{x} = \int_K \nabla u_h \cdot \mathbf{m}_z \, d\mathbf{x} \quad \forall \mathbf{m}_z \in \mathcal{M}_1^2(K) \tag{2.65}$$

Using integration by parts it is possible to continue:

$$\int_K \nabla u_h \cdot \mathbf{m}_z \, d\mathbf{x} = -\int_K u_h \mathrm{div}(\mathbf{m}_z) \, dK + \int_{\partial K} u_h \, \mathbf{m}_z \cdot \mathbf{n} \, ds \quad \forall m_z \in \mathcal{M}_1^2(K) \tag{2.66}$$

where the first integral is computable thanks to $Dv3$ and the second one thanks to $Dv1$ and $Dv2$ degrees of freedom. In matrix form it can be written:

$$\mathbf{Ms} = \mathbf{r} \tag{2.67}$$

where $\mathbf{M}_{zj} = \int_K \mathbf{m}_z \cdot \mathbf{m}_j \, d\mathbf{x}$, $\mathbf{r}_z = \int_K \nabla u_h \cdot \mathbf{m}_z \, d\mathbf{x}$ and $\mathbf{s}_j = s_j$.
Once the projection $\mathbf{\Pi}_1^{0,K} \nabla u_h$ is computed respect to the polynomial basis, the norm (so the integral) in (2.61) are calculated using bidimensional Simpson formula (see (2.58)) if ∇u is known.

Error plot The error is plotted usually in *loglog* scale, in this way the error is a line of slope 2, if plotted respect to geometrical dimension of mesh h, or of slope -1, if plotted respect to the global degrees of freedom number N_{dofs}.

2.5 A posteriori error analysis and mesh refinement

In this section the method used to estimate the error for every element of the mesh is described. The a posteriori error analysis builds an error estimate using the virtual element discrete solution obtained using a given mesh and the data of the problem. In this way it is possible to identify which are the elements where most of error is. So, having marked those elements, it is possible to proceed to refine each one of them. The whole procedure can be described as *adaptive mesh refinement*.

While the a posteriori estimator construction is related to the considered problem, the refinement algorithm for an element of the mesh is a general topic regarding the data management of the mesh itself. So the following description of the algorithm will be valid for every problem where refinement is taken into account.

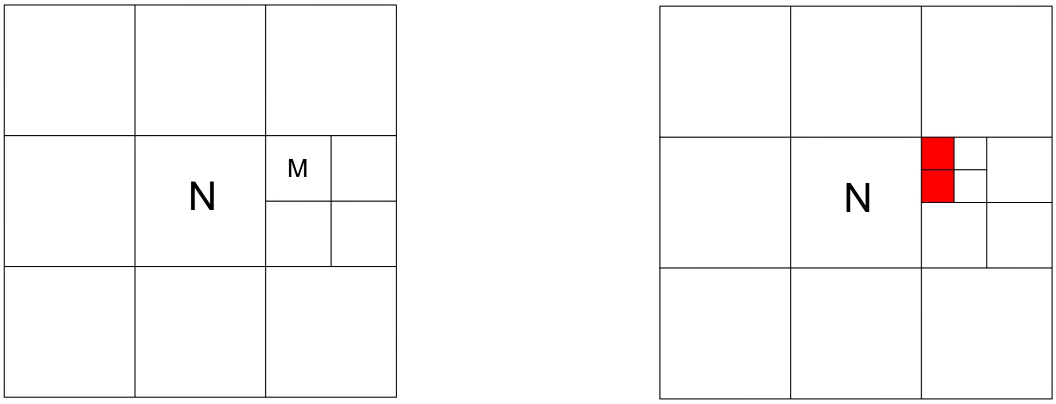
2.5.1 Refinement of a mesh element

The refinement of a single element is done by the function *refine_el* which receives as input the actual mesh and the index of the element which has to be refined (the *marked element*). The output is a new refined mesh.

The refinement depends on how data mesh are organized and saved. In literature an example of data structures used for adaptive mesh refinement of quadrangular meshes are *quadtrees* (see Popinet [2015] and Popinet [2003]). Here a different approach is adopted based on the "observation" of neighbouring elements of the marked one. The resulting algorithm is not so complex thanks to Assumption 2, which implies at most two neighbours per edge, and the way data are saved (hanging nodes are distinguished from other nodes).

The refinement of an element consists in splitting it in four equal parts, namely the four rectangles which come from connecting the midpoints of each *geometrical edge* with the centroid. Therefore from one element four new ones are generated.

A "level of refinement" is assigned to every element choosing that all elements of the first mesh have a level equal to zero. If the marked element has a level of refinement equal to $l+1$ while at least one of its neighbour has the level l, after the refinement one of new elements would have a level of $l+2$ violating the constrain in Assumption 2 (see Figure 2.4). Therefore the neighbouring element must be refined such that its level becomes $l+1$.

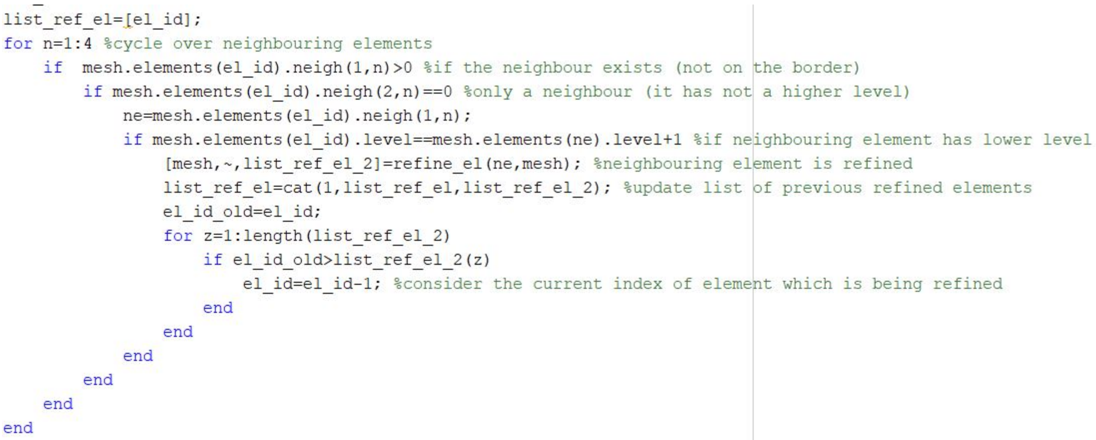


(a) Before refinement

(b) After refinement

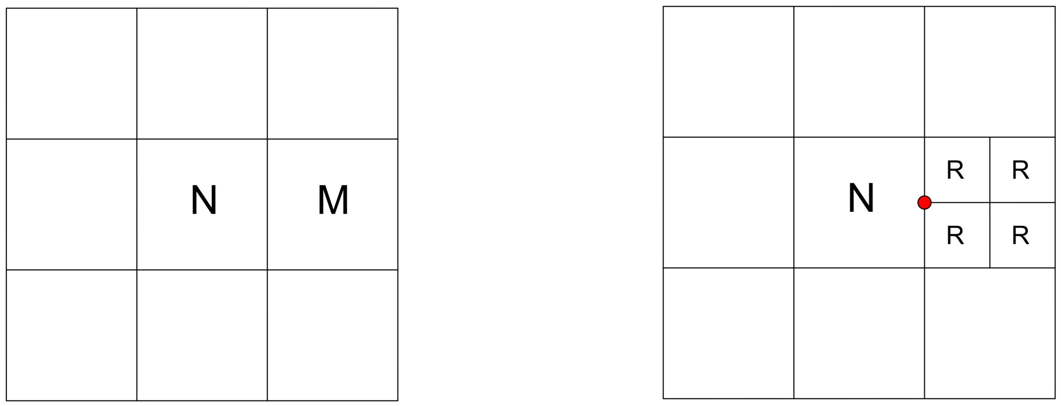
Figure 2.4: M element level is greater than the one of neighbouring element N. After refinement of M new red elements violate Assumption 2 (the difference between their level and N level is equal to 2)

This leads the function *refine_el* to be recursive. In Figure 2.5 the part of the code that checks the level of neighbours is shown. After this first step there are no possibilities of



```
list_ref_el=[el_id];
for n=1:4 %cycle over neighbouring elements
    if  mesh.elements(el_id).neigh(1,n)>0 %if the neighbour exists (not on the border)
        if mesh.elements(el_id).neigh(2,n)==0 %only a neighbour (it has not a higher level)
            ne=mesh.elements(el_id).neigh(1,n);
            if mesh.elements(el_id).level==mesh.elements(ne).level+1 %if neighbouring element has lower level
                [mesh,~,list_ref_el_2]=refine_el(ne,mesh); %neighbouring element is refined
                list_ref_el=cat(1,list_ref_el,list_ref_el_2); %update list of previous refined elements
                el_id_old=el_id;
                for z=1:length(list_ref_el_2)
                    if el_id_old>list_ref_el_2(z)
                        el_id=el_id-1; %consider the current index of element which is being refined
                    end
                end
            end
        end
    end
end
```

Figure 2.5: Recursive part of the element refinement code



(a) Before refinement

(b) After refinement

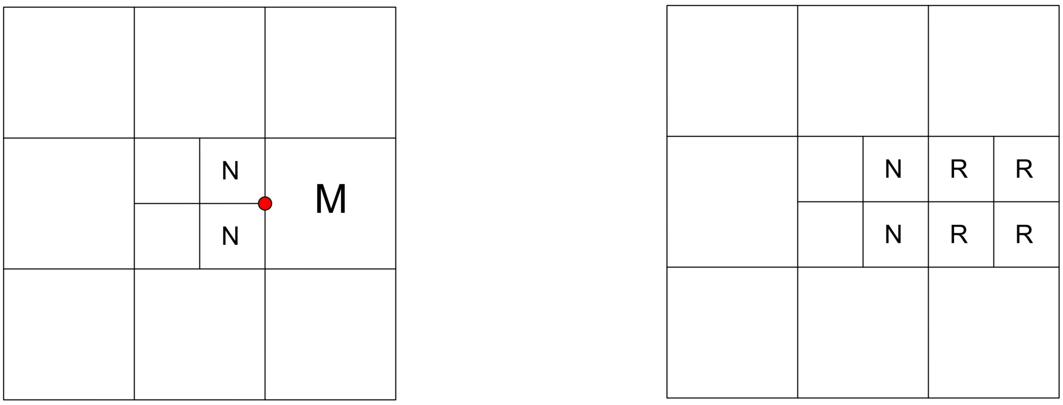
Figure 2.6: In this case, after the refinement of element M, neighbouring element N gets an hanging node (red dot).

violating Assumption 2. So, the new vertices are added starting from the midpoints of the *geometrical edges*. After that the centroid is added. It is important to notice that if one of the *geometrical edges* is composed by two edges (namely there is an hanging node) the related midpoint must not be added because it already exists. For every *geometric edge*, if there is not the hanging node, the previous edge is replaced by two new edges given by the connection between the vertices of previous edge and the midpoint. After that, the edges deriving from the connection between midpoints and centroid are added. Finally the new elements are built in the apposite field.

Now also the neighbours of the marked element must be modified in order to take into account possible hanging nodes and different neighbours (the new elements that has been created through refinement). There are two possible cases:

1. if the level of the original *marked element* is the same of the neighbour the refinement creates a hanging node for the neighbour on the *geometrical edge* shared with the marked element (see Figure 2.6). Therefore the index of the hanging node is added in the correct position and the previous edge is rewritten by the two new ones. Moreover also the indices of neighbouring element of the neighbour (namely the *marked element*) must be changed with the two new born elements from refinement which share an edge with the neighbour.

2. if the level of original *marked element* (l) is lower than the one of the neighbour ($l+1$) (see Figure 2.7) the only thing to do is updating the indices of the neighbouring element of the neighbours (the elements N in Figure 2.7) for the side in common with the marked element.



(a) Before refinement

(b) After refinement

Figure 2.7: In this case, after the refinement of element M, the hanging node (red dot) disappears.

The case of a neighbour of lower level $(l-1)$ with respect to the *marked element* (l) is not taken into account here because it is considered at the beginning doing the recursive refinement.

The last step consists in removing the *marked element* changing coherently the indices of other elements and neighbours.

2.5.2 A Posteriori error analysis for the diffusion problem

The a posteriori error estimate used here for diffusion problem comes from the work done in Cangiani et al. [2017]. In that article an a posteriori estimate for diffusion-convection-reaction problem is derived. Here the upper bound theorem for the error is reported considering only diffusion with diffusion coefficient constant and equal to 1.

Let now R_K be the residual on a element K:

$$R_K = (f_h + \nabla \cdot \mathbf{\Pi}_1^0 \nabla u_h) \tag{2.68}$$

f_h is the approximation of f, already defined in (2.29), and $\mathbf{\Pi}_1^0$ is L^2-projection for vector-valued functions on $[\mathbb{P}_1]^2$ used also for H^1 error definition (2.62). It is important to underline that f_h, here, is constant while in Cangiani et al. [2017] $f_h \in \mathbb{P}_1$.

The residual J_s on an edge s is defined as:

$$J_s = [\![\mathbf{\Pi}_1^0 \nabla u_h]\!]_s \tag{2.69}$$

$[\![\cdot]\!]_s$ indicates the jump of a certain function on edge s. Let K^+ and K^- be the two elements having in common the edge s $(s \in \partial K^+ \cap \partial K^-)$ and let $\mathbf{v}$ be a vector-valued function with trace on s indicated by $\mathbf{v}^{\pm}$. Defined with $\mathbf{n}_s^+$ ($\mathbf{n}_s^-$) the unit outward normal from K^+ (K^-), for every s not belonging to the boundary, $[\![\mathbf{v}]\!]|_s = \mathbf{v}^+ \cdot \mathbf{n}_s^+ + \mathbf{v}^- \cdot \mathbf{n}_s^-$.

If s belongs to the boundary the jump is not calculated (namely it is set to *zero*). This choice is different from the one in Cangiani et al. [2017] where $[\![\mathbf{v}]\!]|_s = \mathbf{v}^+ \cdot \mathbf{n}^+$ because using Dirichlet conditions the quantity $\mathbf{v}^+ \cdot \mathbf{n}^+$ ($\mathbf{v}$ is the gradient in this case) cannot be compared with another quantity on the same edge s.

Indicating with h_s the length of edge s and with h_K the diameter of element K the following theorem, adapted to this case, about the **upper bound** of the error can be proven (for the proof see Cangiani et al. [2017]).

Theorem 3 *Let $u_h \in V_h$ be the discrete virtual element solution of problem (2.8). Then there exists a constant C, independent of h (discrete problem characteristic dimension), u and u_h, such that:*

$$|u - u_h|_1^2 \leq C \sum_{K \in \mathcal{T}_h} (\eta^K + \Theta^K + \Sigma^K) \tag{2.70}$$

where:

$$\eta^K = h_K^2 ||R_K||_{0,K}^2 + \sum_{s \subset \partial K} h_s ||J_s||_{0,s}^2 \tag{2.71}$$

$$\Theta^K = h_K^2 ||f - f_h||_{0,K}^2 \tag{2.72}$$

Σ^K is a term which depends on the stabilizing form S^K:

$$\Sigma^K = S^K((\Pi_2^0 - I)u_h, (\Pi_2^0 - I)u_h) \tag{2.73}$$

where I is the identity.

$|| \cdot ||_{0,K}$ is L^2 norm on the element K and $|| \cdot ||_{0,s}$ is L^2 norm on edge s. Π_2^0 is L^2-projector on $\mathbb{P}_2$.

The following result found in Beirão da Veiga et al. [2021b] is now reported.

Proposition 4 *The inequality (2.70) can be rewritten changing constant C*

$$|u - u_h|_1^2 \leq D \sum_{K \in \mathcal{T}_h} (\eta^K + \Theta^K) \tag{2.74}$$

where D is a new constant independent of h.

Moreover it is defined:

$$P^K = \eta^K + \Theta^K \tag{2.75}$$

Remark 7 *The Proposition 4 implies that Σ^K does not affect the error estimate. In fact Σ^K can be controlled by η^K as shown in the recent work Beirão da Veiga et al. [2021b].*

Remark 8 *The a posteriori error was built using a non-enhanced Virtual element space differently from the work in Cangiani et al. [2017]. The use of the enhanced Virtual element space allows to build the L^2 projection on polynomials of degree 2 ($\mathbb{P}_2$). Practically this computation is necessary for the estimation (Σ^K) coming from stabilization term that, here, was not included as already explained.*

2.5.3 Solve-Estimate-Mark-Refine

As explained in Cangiani et al. [2017], the procedure adopted in adaptive mesh refinement is the so called **Solve-Estimate-Mark-Refine**. The first step *solve* consists in solving the VEM problem for a certain mesh (and so for a certain decomposition $\mathcal{T}_h$). Then for every element $K \in \mathcal{T}_h$ a posteriori error estimate is computed (*estimate*).

After that, elements with the largest estimators are *marked*. To decide how many and which elements to mark it is necessary to define a subset $\tilde{\mathcal{T}}_h \subset \mathcal{T}_h$ with minimal cardinality such that:

$$\left(\sum_{K \in \tilde{\mathcal{T}}_h} P^K \right) \geq \theta \left(\sum_{K \in \mathcal{T}_h} P^K \right) \tag{2.76}$$

where P^K is the a posteriori error estimator for element K (see (2.75)) and θ is a parameter which belongs to interval (0,1). (2.76) represents Dörfler/bulk marking strategy (Cangiani et al. [2017]). In Cangiani et al. [2017] $\theta = 0.4$, here, for each case, the value of θ will be specified. Practically the elements, listed in descending order of estimators, are marked starting from the one with largest estimator. The procedure is stopped when the threshold described in (2.76) is reached. In other words the marked set contains a part of the total a posteriori estimate equal to θ.

The last step (*refine*) refines the marked elements following the algorithm described above. So at the end the problem can be solved on a new refined mesh and the procedure described can be repeated. Obviously the procedure can create meshes where there are hanging nodes.

2.6 Numerical tests

Here numerical results, obtained through the VEM technology which has been implemented, are reported. Tests presented here have not a physical importance but they want to show how the VEM described works. The section is organized in two subsections: the first one shows cases where each element of the mesh has been refined at each step, while the second one shows cases where adaptive mesh refinement is adopted. Both of them contain plots of solution gradient error and of a posteriori estimators as described in previous parts. Moreover they are plotted versus the number of degrees of freedom N_{dofs} Some plots of solution function obtained by VEM are shown using *Patch* (MathWorks) function of *Matlab* as done in Sutton [2017]. In errors plots legend "gradient error" refers to error defined in (2.61), while "a posteriori estimator" is obtained from P^K defined in (2.75). In other words it is given by:

$$\left(\sum_{K \in \mathcal{T}_h} (\eta^K + \Theta^K) \right)^{1/2} \tag{2.77}$$

Moreover another plot is realized separating different contributions to the a posteriori estimator. Using names reported in legend, Θ is:

$$\left(\sum_{K\in\mathcal{T}_h}(\Theta^K)\right)^{1/2} \tag{2.78}$$

where Θ^K is given by (2.72).

η is:

$$\left(\sum_{K\in\mathcal{T}_h}(\eta^K)\right)^{1/2} \tag{2.79}$$

where η^K is given by (2.71).

Moreover element error part and error coming from "jumps" on edges in (2.71) are shown separately. η_no_edges means:

$$\left(\sum_{K\in\mathcal{T}_h}(h_K^2||R_K||_{0,K}^2)\right)^{1/2} \tag{2.80}$$

η_edges is

$$\left(\sum_{K\in\mathcal{T}_h}\sum_{s\subset\partial K} h_s||J_s||_{0,s}^2\right)^{1/2} \tag{2.81}$$

Furthermore tables containing gradient errors and a posteriori estimators are reported. The mesh used in all those case has parameters $A = 6$ and $L = 10$ (defined in Geometry section).

2.6.1 Uniform mesh refinements

In a uniform mesh refinement at every step elements have all the same size and shape, in other words there are no hanging nodes. For all cases refinements are done starting from the coarsest mesh whose elements have one edge equal to 1/2 of square cylinder side (which is 1) length.

Case 1 The starting mesh here is shown in Figure 2.8. The function solution in this case is:

$$u = \frac{1}{100}(-6-x)(10-x)(-4-y)(4-y) \tag{2.82}$$

and the right-hand side term f is computed from u. On the external boundary there are homogeneous Dirichlet boundary conditions while on the obstacle boundaries non-homogeneous Dirichlet boundary conditions with $g = u$ are imposed.

3 refinements are realized. Figure 2.9 shows plot of VEM discrete solution for the second refinement. Table 2.4 shows gradient errors and a posteriori estimators. Figure 2.10a shows gradient errors and a posteriori estimators and Figure 2.10b shows various contributions in a posteriori estimator.

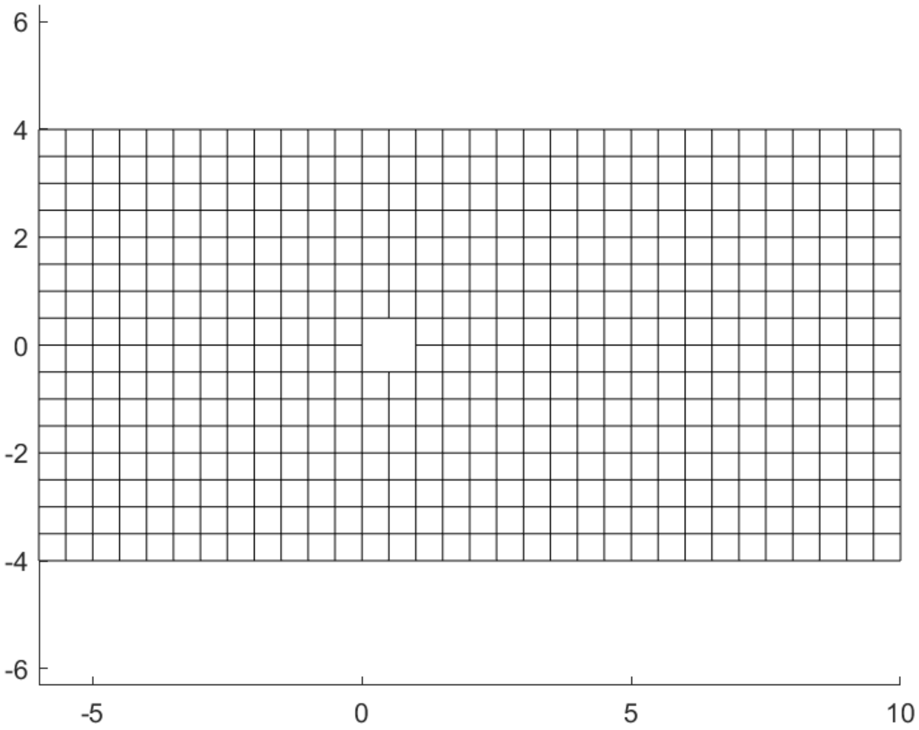


Figure 2.8: Starting mesh with $A = 6$ and $L = 10$ (coarsest mesh)

h	H^1 error (gradient error)	a posteriori estimator
1/2	0.0665985	0.332131497
1/4	0.0166640	0.0836847977
1/8	0.0041670	0.0209989936
1/16	0.0010418	0.00525924272

Table 2.4: (Diffusion) Gradient errors and a posteriori estimators for uniform refinement sequence: *case 1*

It is possible to notice that the gradient error follows theoretical results. The a posteriori estimator is parallel to the true error and about one order of magnitude greater as in results of Cangiani et al. [2017]. The greater contributions in a posteriori estimator are Θ and η_edges.

Case 2 Data are chosen in order to have as function solution:

$$u = \cos(\pi y) \sin(\pi x) \tag{2.83}$$

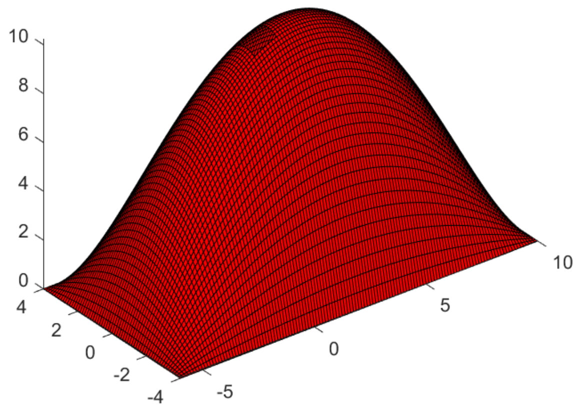
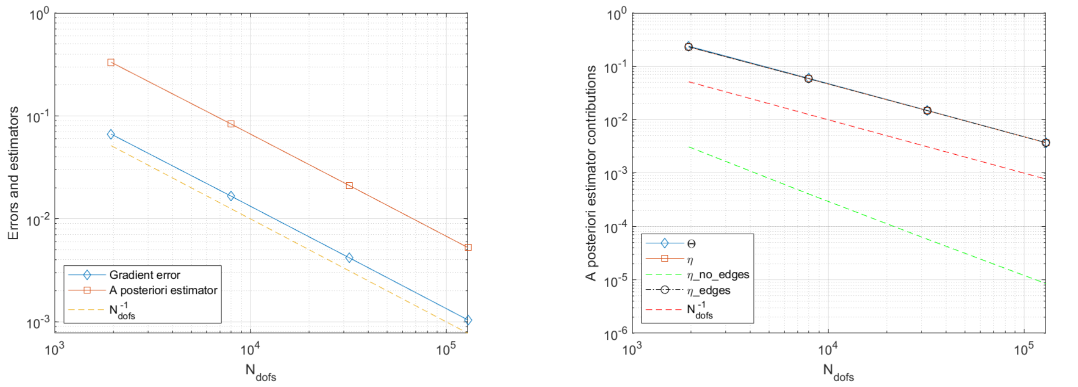


Figure 2.9: (Diffusion) Plot of VEM solution of *case 1* (with mesh of second refinement)



(a) Plot of error and estimator

(b) Plot of parts of a posteriori estimator

Figure 2.10: (Diffusion) *Case 1* errors and estimators

$f = 2\pi^2(\cos(\pi y)\sin(\pi x))$. On the obstacle boundary there are homogeneous Dirichlet conditions while on the external boundary non-homogeneous Dirichlet boundary conditions with $g = u$ are imposed. The starting mesh here is shown in Figure 2.8. 2 refinements are realized. Figure 2.11 shows plot of VEM discrete solution for the second refinement. Table 2.5 shows gradient errors and a posteriori estimators. Figure 2.12a shows gradient and a posteriori estimators and Figure 2.12b shows various contributions in a posteriori estimator.

Also in this case errors follows theoretical results. Θ is the most important contribution in a posteriori estimator.

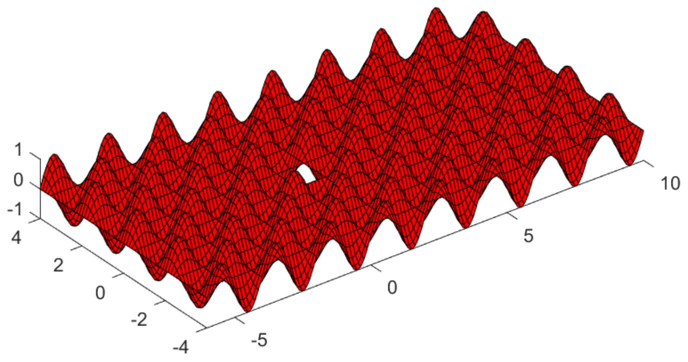
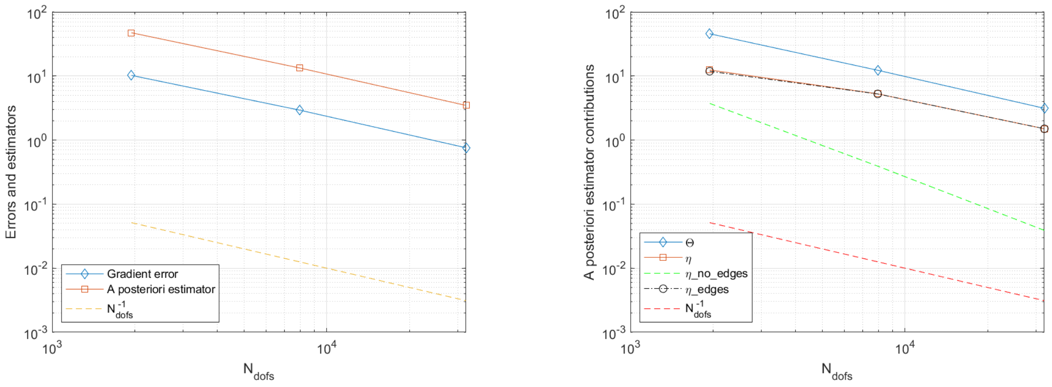


Figure 2.11: (Diffusion) Plot of VEM solution of *case 2* (with mesh of second refinement)

h	H^1 error (gradient error)	a posteriori estimator
1/2	10.226717417	47.31264374565
1/4	2.9363746573	13.3713129574
1/8	0.75850238587	3.4736326507

Table 2.5: (Diffusion) Gradient errors and a posteriori estimators for uniform refinement sequence: *case 2*



(a) Plot of error and estimator

(b) Plot of parts of a posteriori estimator

Figure 2.12: (Diffusion) *Case 2* errors and estimators

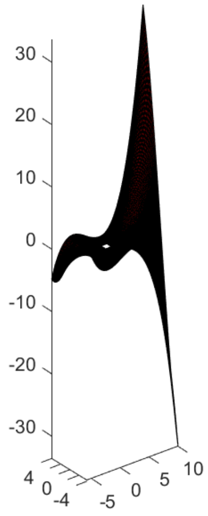


Figure 2.13: (Diffusion) Plot of VEM solution of *case 3* (with mesh of second refinement)

h	H^1 error (gradient error)	a posteriori estimator
1/2	0.110693099331365	0.364097771896506
1/4	0.0276837768347681	0.0934667596327699
1/8	0.00692160042820237	0.0236722134103287

Table 2.6: (Diffusion) Gradient errors and a posteriori estimators for uniform refinement sequence: *case 3*

Case 3 Data are chosen in order to have as function solution:

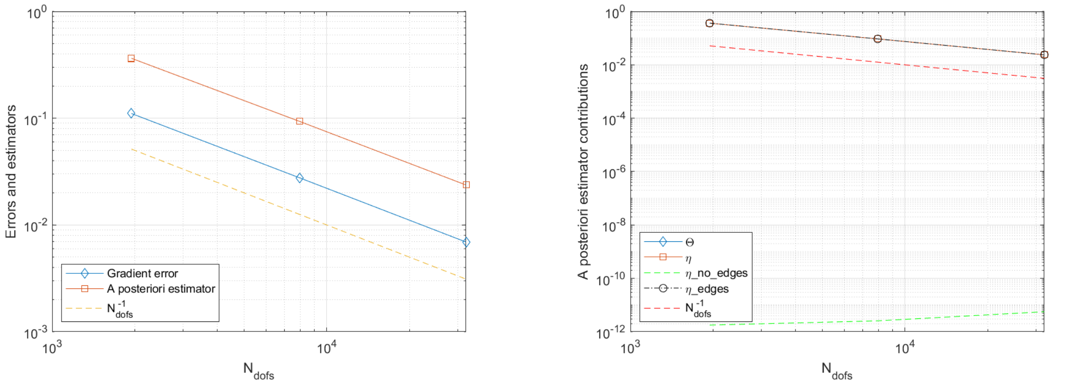
$$u = \frac{1}{100}(yx^3 - y^3x) \tag{2.84}$$

In this case $f = 0$. On all boundaries non-homogeneous Dirichlet conditions with $g = u$ are imposed. The starting mesh here is shown in Figure 2.8. 2 refinements are realized. Figure 2.13 shows plot of VEM discrete solution for the second refinement. Table 2.6 shows gradient errors and a posteriori estimators. Figure 2.14a shows gradient errors and a posteriori estimators and Figure 2.14b shows various contributions in a posteriori estimator.

Errors and a posteriori estimators follow classical trend. In this case Θ is zero because $f = 0$. The main contribution for a posteriori estimator is given by η, in particular η_edges.

Case 4 Data are chosen in order to have as function solution:

$$u = 4e^{(-((x-4)^2+y^2)/2)} \tag{2.85}$$



(a) Plot of error and estimator

(b) Plot of parts of a posteriori estimator

Figure 2.14: (Diffusion) *Case 3* errors and estimators

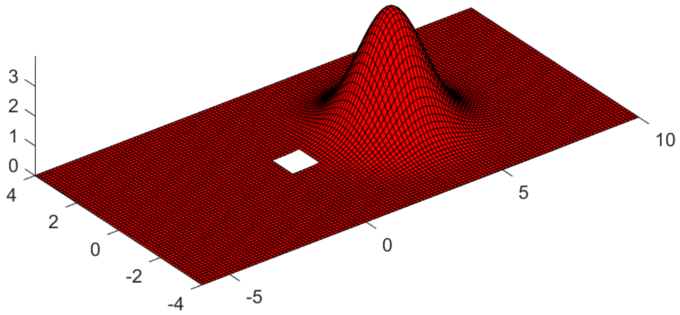


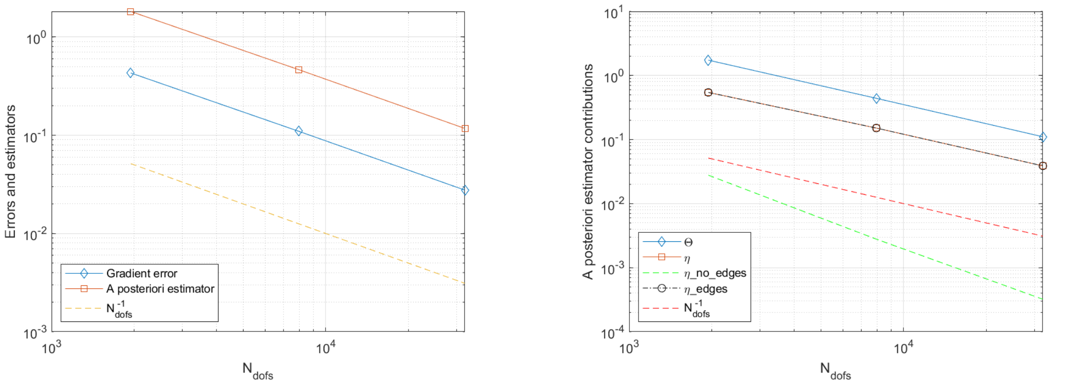
Figure 2.15: (Diffusion) Plot of VEM solution of *case 4* (with mesh of second refinement)

So the solution in this case is a Gaussian. On all boundaries non-homogeneous Dirichlet conditions with $g = u$ are imposed. The starting mesh here is shown in Figure 2.8. 2 refinements are realized. Figure 2.15 shows plot of VEM discrete solution for the second refinement. Table 2.7 shows gradient errors and a posteriori estimators. Figure 2.16a shows gradient errors and a posteriori estimators and Figure 2.16b shows various contributions in a posteriori estimator.

Errors confirms theoretical results and a posteriori estimator is parallel to the true error and about 1 order of magnitude greater. The main contribution for a posteriori estimator is given by Θ.

h	H^1 error (gradient error)	a posteriori estimator
1/2	0.430497527819063	1.81956078215340
1/4	0.109946964615904	0.465984434366007
1/8	0.027634508927050	0.117229658267013

Table 2.7: (Diffusion) Gradient errors and a posteriori estimators for uniform refinement sequence: *case 4*



(a) Plot of error and estimator

(b) Plot of parts of a posteriori estimator

Figure 2.16: (Diffusion) *Case 4* errors and estimators

h	H^1 error (gradient error)	a posteriori estimator
1/2	0.283572301529935	1.12484659052553
1/4	0.0718889456395198	0.285297579777855
1/8	0.0180351654066124	0.0715957250842429

Table 2.8: (Diffusion) Gradient errors and a posteriori estimators for uniform refinement sequence: *case 5*

Case 5 Data are chosen in order to have as function solution:

$$u = 4e^{(-((x-2)^2+y^2)/2)} \tag{2.86}$$

So the solution in this case is similar to *Case 4* but the Gaussian is shifted to left. On all boundaries non-homogeneous Dirichlet conditions with $g = u$ are imposed. The starting mesh here has rectangular elements and it is shown in Figure 2.17. So elements are rectangles where vertical edge has double length with respect to horizontal edge. 2 refinements are realized. Figure 2.18 shows plot of VEM discrete solution for the second refinement. Table 2.8 shows gradient errors and a posteriori estimators. Figure 2.19a shows gradient errors and a posteriori estimators and Figure 2.19b shows various contributions in a posteriori estimator.

Also in this case errors and a posteriori estimators follow the usual trend. Θ is the

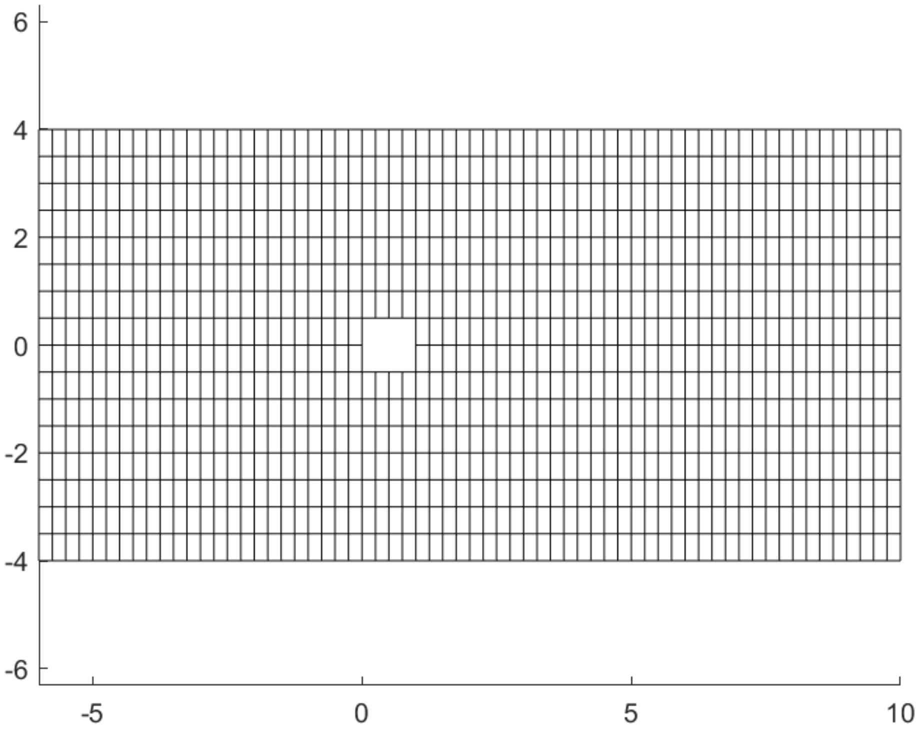


Figure 2.17: Starting mesh with $A = 6$ and $L = 10$ (coarsest mesh)

main contribution in a posteriori estimator.

2.6.2 Adaptive mesh refinements

Adaptive mesh refinement is done following *solve-estimate-mark-refine* procedure described in previous parts. Cases considered are the same as uniform mesh refinements, therefore considerations about the data are not reported again. The number of refinements is specified for every case and some meshes, with hanging nodes, coming from resolution are shown.

Case 1 In this case parameter θ is chosen $\theta = 0.5$. The starting mesh is the same as uniform case and it is refined 11 times. Figure 2.20 shows plot of the VEM discrete solution using the mesh obtained with last refinement. Figure 2.22a shows gradient errors and a posteriori estimators. Figure 2.22b shows various contributions in a posteriori estimator. Figure 2.21 shows the sequence of meshes in this case. Error confirms theoretical results and a posteriori estimator follows the trend of error. The top part of function is the last one which is refined (see Figure 2.21).

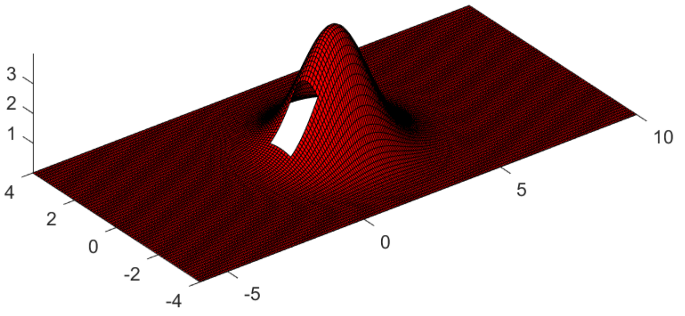
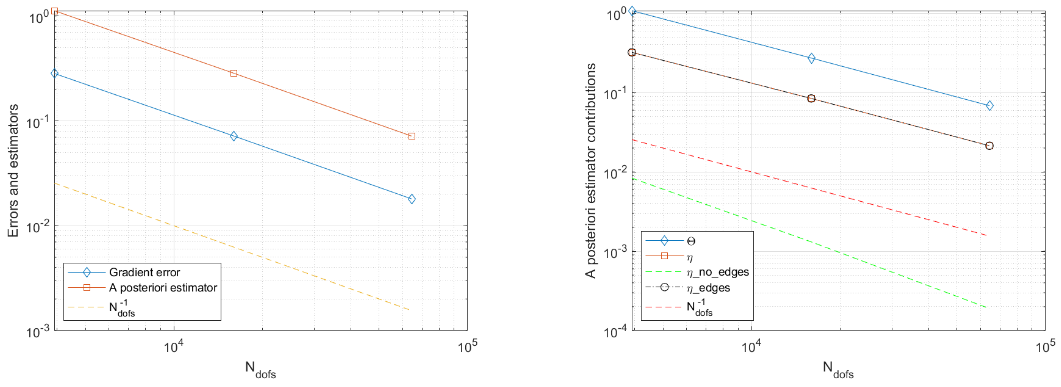


Figure 2.18: (Diffusion) Plot of VEM solution of *case 5* (with mesh of second refinement)



(a) Plot of error and estimator

(b) Plot of parts of a posteriori estimator

Figure 2.19: (Diffusion) *Case 5* errors and estimators

Case 2 In this case parameter θ is chosen $\theta = 0.4$. The starting mesh is the same as uniform case and it is refined 15 times. Figure 2.23 shows plot of the VEM discrete solution using the mesh obtained with last refinement. Figure 2.25a shows gradient errors and a posteriori estimators and Figure 2.25b shows various contributions in a posteriori estimator. Figure 2.24 shows the sequence of meshes. Also in this case errors and a posteriori estimators follow classical trends. Last meshes (see Figure 2.24) show that refinement is similar in every oscillation.

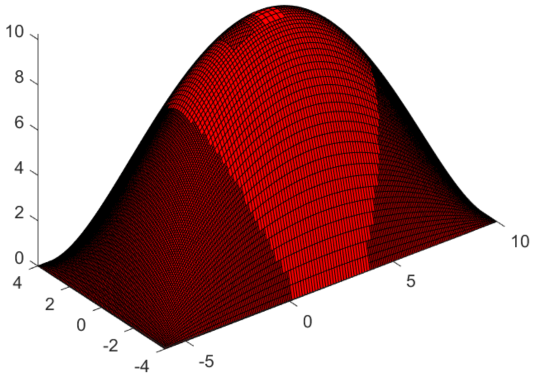
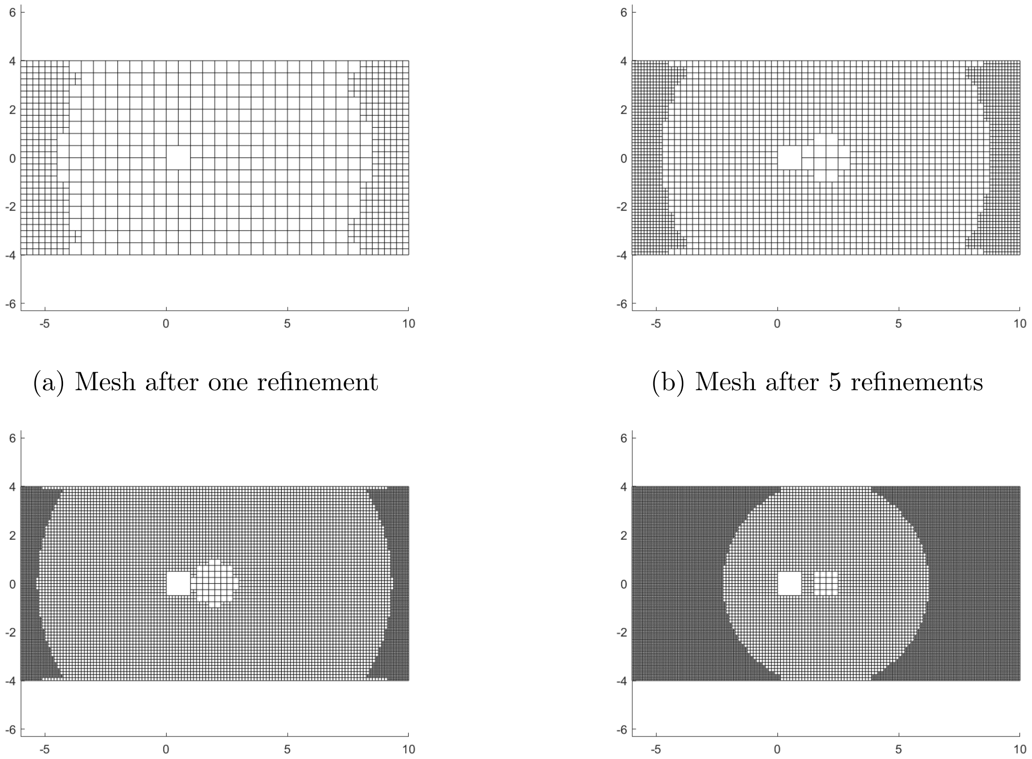


Figure 2.20: (Diffusion) Plot of VEM solution of *case 1* (adaptive) after 11 refinements

Case 3 In this case parameter θ is chosen $\theta = 0.4$. The starting mesh is the same as uniform case and it is refined 15 times. Figure 2.26 shows plot of the VEM discrete solution using the mesh obtained with last refinement. Figure 2.28a shows gradient errors and a posteriori estimators. Figure 2.28b shows various contributions in a posteriori estimator. Figure 2.27 shows the sequence of meshes. Also in this case the error and the a posteriori estimator follow a normal trend. For what concerns the a posteriori estimator the main contribution is given by η_edges as shown in Figure 2.28b.

Case 4 In this case parameter θ is chosen $\theta = 0.4$. The starting mesh is the same as uniform case and it is refined 15 times. Figure 2.29 shows plot of VEM discrete solution using the mesh obtained with last refinement. Figure 2.31a shows gradient errors and a posteriori estimators. Figure 2.31b shows various contribution in a posteriori estimator. Figure 2.30 shows the sequence of meshes. Again errors and a posteriori estimators follow classical trends. Moreover the Gaussian shows well (see 2.30) that a posteriori estimator refines the part of mesh where the function is "located".

Remark 9 *It is important to notice that in all cases, even if stabilization part has not been included in a posteriori estimator, numerical results make sense confirming what was found in Beirão da Veiga et al. [2021b].*



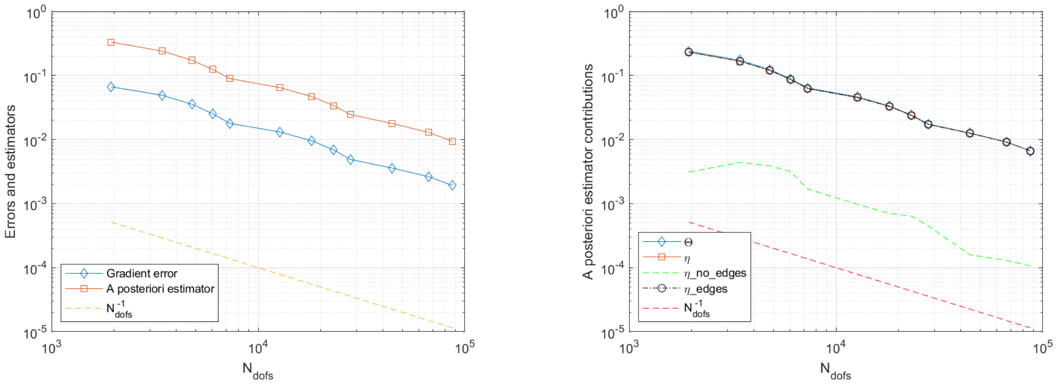
(a) Mesh after one refinement

(b) Mesh after 5 refinements

(c) Mesh after 9 refinements

(d) Mesh after 11 refinements

Figure 2.21: (Diffusion) 4 meshes generated in adaptive mesh refinement of *case 1*



(a) Plot of error and estimator

(b) Plot of parts of a posteriori estimator

Figure 2.22: (Diffusion) *Case 1* (adaptive) errors and estimators

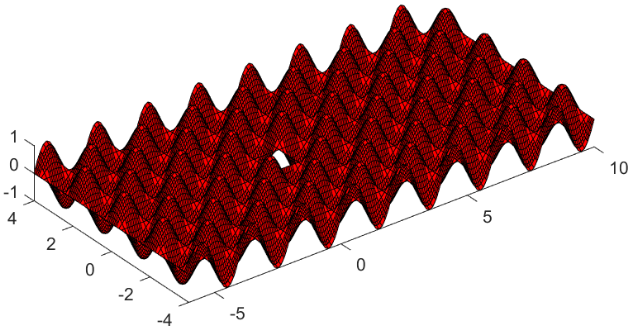
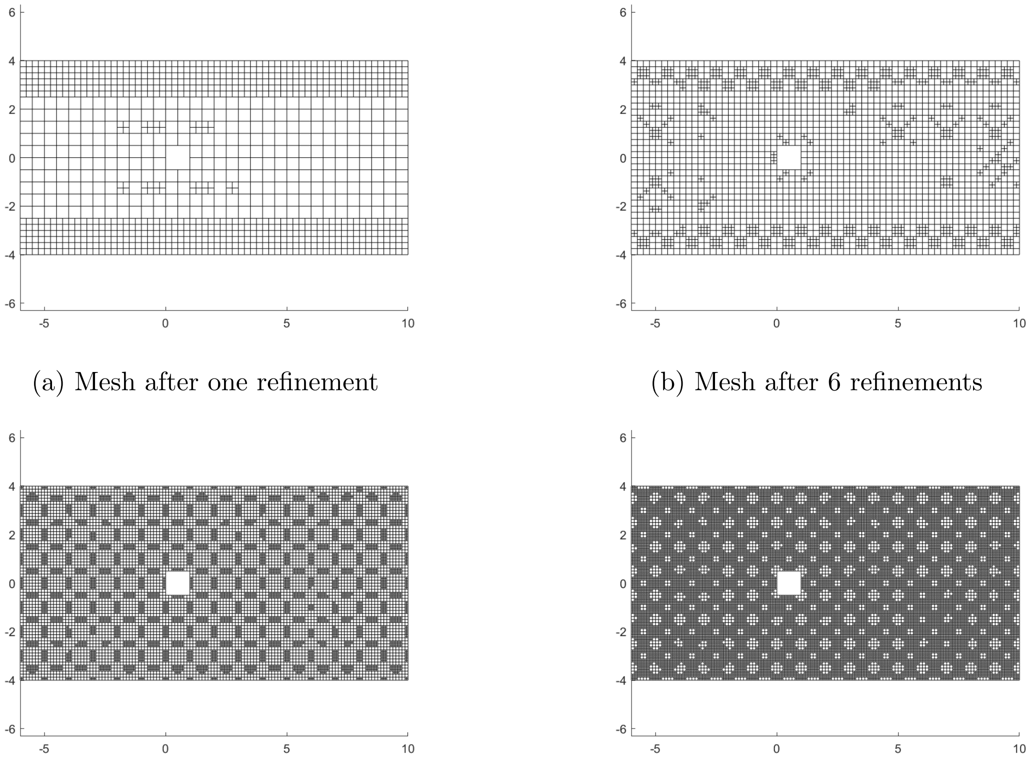


Figure 2.23: (Diffusion) Plot of VEM solution of *case 2* (adaptive) after 15 refinements



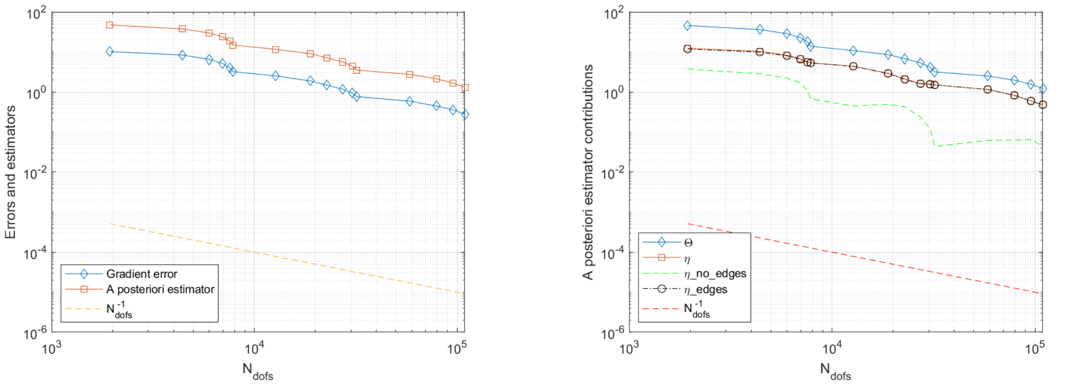
(a) Mesh after one refinement

(b) Mesh after 6 refinements

(c) Mesh after 12 refinements

(d) Mesh after 15 refinements

Figure 2.24: (Diffusion) 4 meshes generated in adaptive mesh refinement of *case 2*



(a) Plot of error and estimator

(b) Plot of parts of a posteriori estimator

Figure 2.25: (Diffusion) *Case 2* (adaptive) errors and estimators

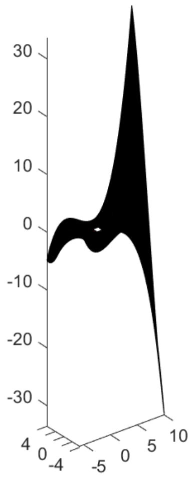
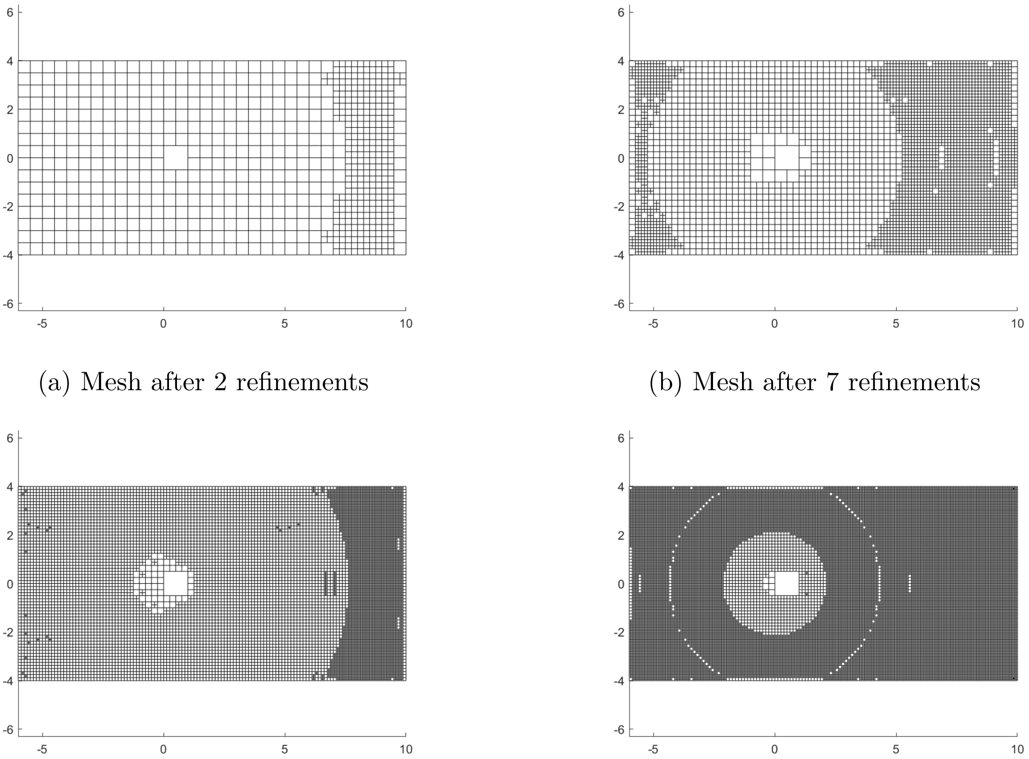


Figure 2.26: (Diffusion) Plot of VEM solution of *case 3* (adaptive) after 15 refinements



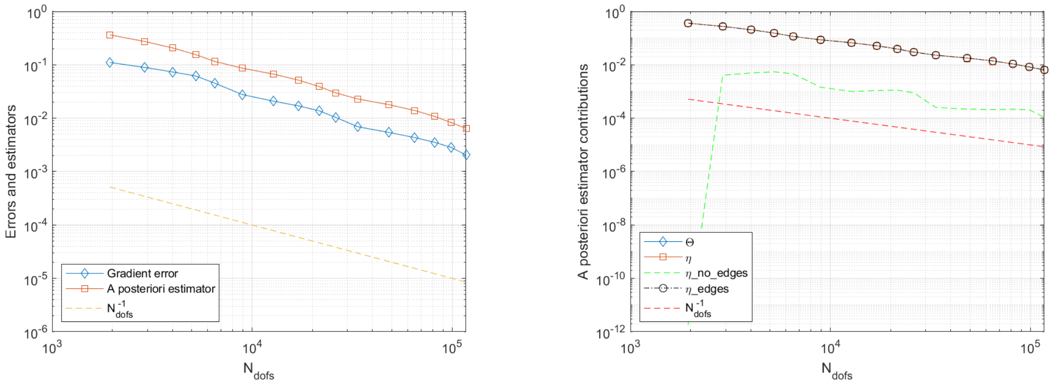
(a) Mesh after 2 refinements

(b) Mesh after 7 refinements

(c) Mesh after 11 refinements

(d) Mesh after 15 refinements

Figure 2.27: (Diffusion) 4 meshes generated in adaptive mesh refinement of *case 3*



(a) Plot of error and estimator

(b) Plot of parts of a posteriori estimator

Figure 2.28: (Diffusion) *Case 3* (adaptive) errors and estimators

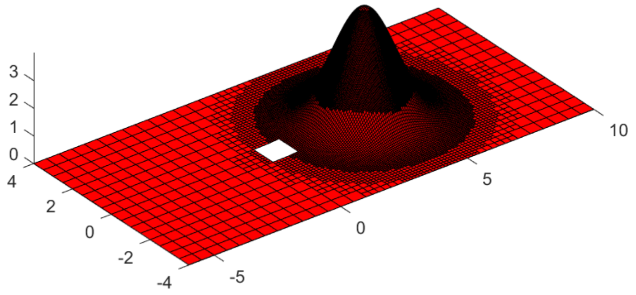
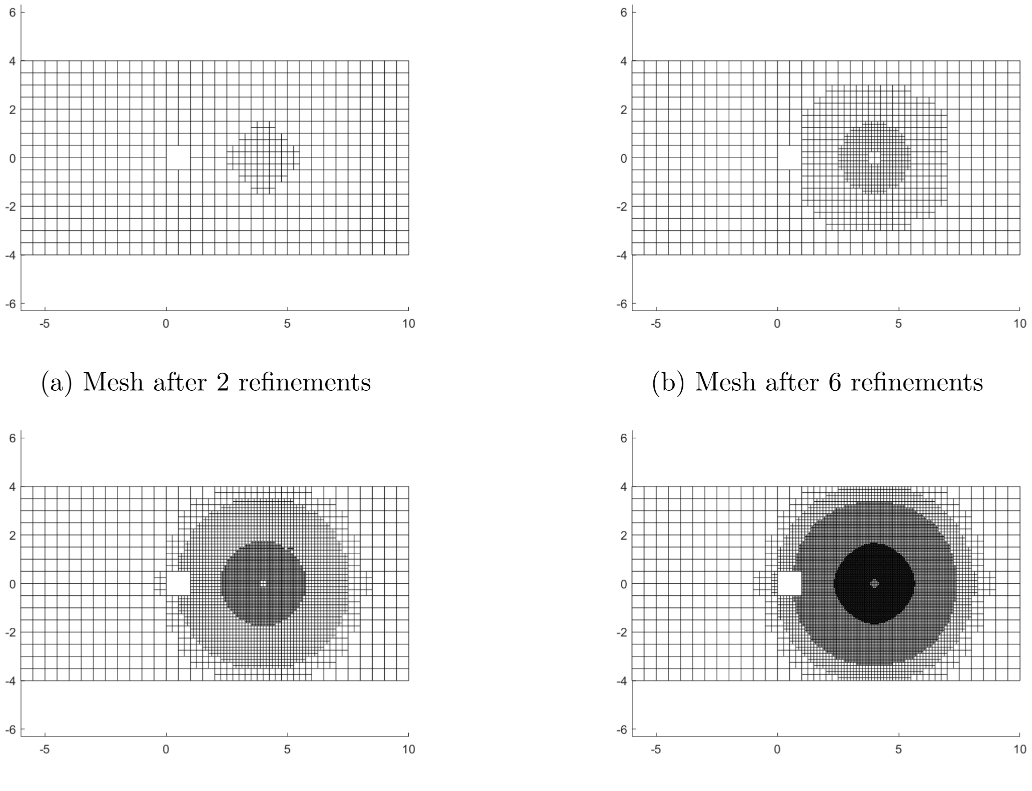


Figure 2.29: (Diffusion) Plot of VEM solution of *case 4* (adaptive) after 15 refinements



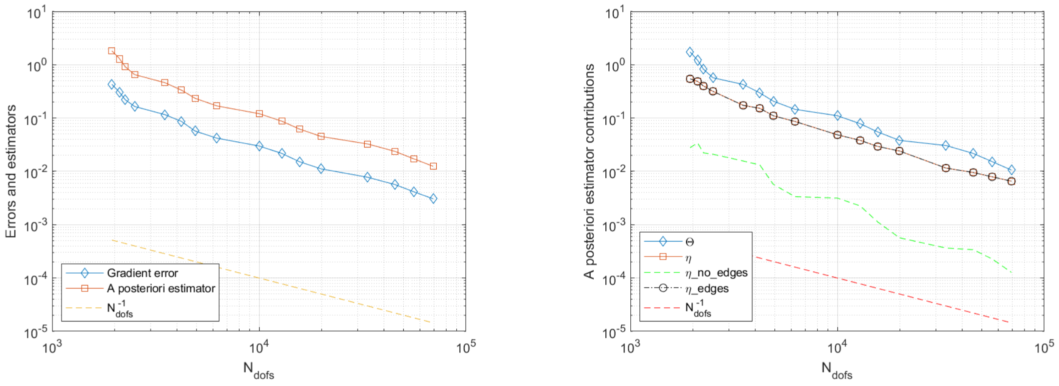
(a) Mesh after 2 refinements

(b) Mesh after 6 refinements

(c) Mesh after 11 refinements

(d) Mesh after 15 refinements

Figure 2.30: (Diffusion) 4 meshes generated in adaptive mesh refinement of *case 4*



(a) Plot of error and estimator

(b) Plot of parts of a posteriori error

Figure 2.31: (Diffusion) *Case 4* (adaptive) errors and estimators

Chapter 3

Solution of the Stokes problem using the VEM

The Stokes equations system describes the motion of an incompressible fluid without considering the convection effects, in fact it can represent real situations where convection is irrelevant with respect to diffusion. Therefore it can be seen as a simplification of the more complex case of Navier-Stokes problem. The whole formulation of the VEM discrete Stokes problem and most of the structure of next two sections is taken from the work in Beirão da Veiga et al. [2017]. All numerical techniques presented have been implemented in *Matlab*.

3.1 The continuous problem

The Stokes problem we consider here is defined on a polygonal domain $\Omega \subset \mathbb{R}^2$ as follows:

$$\text{find} \quad (\mathbf{u}, p) \quad \text{such that} \quad \begin{cases} -\nu \boldsymbol{\Delta} \mathbf{u} - \nabla p = \mathbf{f} & \text{in } \Omega \\ \text{div}\mathbf{u} = 0 \\ \mathbf{u} = \mathbf{0}, \end{cases} \tag{3.1}$$

where $\mathbf{u}$ is the velocity vector field while p is the pressure scalar field. In the Stokes problem written here, homogeneous Dirichlet boundary condition are chosen, however, as done in the implementation, it is possible to extend the VEM formulation of the problem to non-homogeneous Dirichlet boundary conditions and to Neumann conditions. It is also important to underline that the problem is formulated by subtracting the pressure gradient $(-\nabla p)$ as done in Beirão da Veiga et al. [2017] while in other cases (Wang et al. [2020]) the problem is presented with $+\nabla p$. Anyway the two problems are equivalent since the only difference is the sign of pressures (the two pressures have the same absolute value with different sign). ν is the viscosity and it is a constant scalar field while $\mathbf{f}$ is the external force. $\boldsymbol{\Delta}$ is the vector-valued Laplacian ($\boldsymbol{\Delta}[u_x, u_y] = [\Delta u_x, \Delta u_y]$). $\nabla \mathbf{u}$ will indicate the tensor-valued gradient (the Jacobian matrix) of the vector field $\mathbf{u}$. div and ∇ are the divergence and the gradient operators.

General boundary conditions can be written in compact form as (see Beirão da Veiga et al. [2021a]):

$$\begin{cases}\mathbf{u} = \mathbf{g} \quad \text{on} \quad \partial\Omega_D \\ \mathbf{T}(\mathbf{u}, p)\mathbf{n} = (\nu\nabla\mathbf{u} + p\mathbf{I})\mathbf{n} = \mathbf{\Psi} \quad \text{on} \quad \partial\Omega_N\end{cases} \tag{3.2}$$

Non-homogeneous Dirichlet boundary conditions On the Dirichlet boundary, to impose non-homogeneous conditions, it should be written $\mathbf{u} = \mathbf{g}$ where $\mathbf{g}$ is a known vector-valued (two component) function on Dirichlet boundary $\partial\Omega_D$. Non-homogeneous Dirichlet boundary conditions do not change the variational formulation of the continuous problem. However, in this case, additional terms must be considered during implementation as explained later.

Homogeneous Neumann boundary conditions Boundary conditions on Neumann boundary $\partial\Omega_N$ are $(\nu\nabla\mathbf{u} + p\mathbf{I})\mathbf{n} = \mathbf{\Psi}$ where $\mathbf{I} \in \mathbb{R}^{2\times 2}$ is the identity matrix and $\mathbf{n}$ is the unit outward normal vector on $\partial\Omega_N$. $\mathbf{\Psi}$ is a known vector-valued function which, here, will be always equal to $\mathbf{0}$ because homogeneous conditions are chosen.

Neumann boundary conditions add a term $\int_{\partial\Omega_N} \mathbf{\Psi} \cdot \mathbf{n} \, ds$ to the variational formulation of continuous problem. Choosing homogeneous conditions this term vanishes. However Neumann boundary conditions require a proper consideration during implementation as explained later as well as a different space for pressure with respect to the Dirichlet case in the variational problem formulation.

Considering, for simplicity, only homogeneous conditions on the Dirichlet boundary, let now be the space related to velocity field:

$$\mathbf{V} := \{\mathbf{v} \in [H^1(\Omega)]^2 : \mathbf{v} = \mathbf{0} \quad \text{on } \partial\Omega_D\} \tag{3.3}$$

If only Dirichlet conditions are present in the problem (namely $\partial\Omega_N = \emptyset$) the space related to pressure field is defined as:

$$Q := L^2_0(\Omega) = \{q \in L^2(\Omega) \quad \text{s.t.} \quad \int_\Omega q \, \mathrm{d}\Omega = 0\} \tag{3.4}$$

The condition $\int_\Omega q \, \mathrm{d}\Omega$ is due to the fact that in Stokes problem definition there is only the gradient of p. Therefore pressure p is defined up to a constant value. While in case of presence of Neumann conditions ($\partial\Omega_N \neq \emptyset$) the space becomes:

$$Q^N := L^2(\Omega) \tag{3.5}$$

In fact in this case a condition on pressure is given in Neumann boundary conditions as explained above. The following norms for spaces $\mathbf{V}$ and Q (Q^N) are adopted:

$$||\mathbf{v}||_1 := ||\mathbf{v}||_{[H^1_0(\Omega)]^2}, \quad ||q||_{Q(Q^N)} := ||q||_{L^2(\Omega)} \tag{3.6}$$

Following formulation in Beirão da Veiga et al. [2017], it is assumed that $\mathbf{f} \in [H^{-1}(\Omega)]^2$ and $\nu \in L^\infty(\Omega)$ uniformly positive in Ω. Now the following bilinear forms $a(\cdot,\cdot) : \mathbf{V} \times \mathbf{V} \to \mathbb{R}$ and $b(\cdot,\cdot) : \mathbf{V} \times Q \, (Q^N) \to \mathbb{R}$ are defined:

$$a(\mathbf{u}, \mathbf{v}) := \int_\Omega \nu \, \nabla\mathbf{u} : \nabla\mathbf{v} \, \mathrm{d}\Omega, \quad \forall \, \mathbf{u}, \mathbf{v} \in \mathbf{V} \tag{3.7}$$

$$b(\mathbf{v}, q) := \int_{\Omega} \operatorname{div}(\mathbf{v})\, q \, \mathrm{d}\Omega \quad \forall \mathbf{v} \in \mathbf{V}, q \in Q\,(Q^N) \tag{3.8}$$

In this way it is possible to write the variational Stokes problem:

$$\text{find } (\mathbf{u}, p) \in \mathbf{V} \times Q\,(Q^N) \text{ such that } \quad \begin{cases} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = (\mathbf{f}, \mathbf{v}) & \forall\, \mathbf{v} \in \mathbf{V} \\ b(\mathbf{u}, q) = 0 \quad \forall\, q \in Q\,(Q^N) \end{cases} \tag{3.9}$$

It is possible to show that (see Beirão da Veiga et al. [2017]):

1. $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ are continuous:

$$|a(\mathbf{u}, \mathbf{v})| \leq ||a||||\mathbf{u}||_1||\mathbf{v}||_1 \quad \forall \mathbf{u}, \mathbf{v} \in \mathbf{V}, \tag{3.10}$$

$$|b(\mathbf{u}, q)| \leq ||b||||\mathbf{v}||_1||\mathbf{q}||_1 \quad \forall \mathbf{v} \in \mathbf{V}, \quad \forall q \in Q\,(Q^N) \tag{3.11}$$

2. $a(\cdot, \cdot)$ is coercive: there exists a positive constant α such that

$$a(\mathbf{v}, \mathbf{v}) \geq \alpha||\mathbf{v}||_1^2 \quad \forall \mathbf{v} \in \mathbf{V} \tag{3.12}$$

3. $b(\cdot, \cdot)$ satisfies *inf-sup condition*:

$$\exists \beta > 0 \quad \text{such that} \quad \sup_{\mathbf{v} \in \mathbf{V}, \mathbf{v} \neq \mathbf{0}} \frac{b(\mathbf{u}, q)}{||\mathbf{v}||_1} \geq \beta ||q||_Q \quad \forall q \in Q\,(Q^N) \tag{3.13}$$

Under this condition it is known that the variational Stokes problem has a unique solution $(\mathbf{u}, p) \in \mathbf{V} \times Q\,(Q^N)$ such that:

$$||\mathbf{u}||_1 + ||p||_Q \leq C||\mathbf{f}||_{H^{-1}(\Omega)} \tag{3.14}$$

where C is a constant which depends only on Ω and ν.

3.2 VEM Discrete Stokes problem formulation

3.2.1 Discretization

Degrees of freedom and VEM spaces

As done for the diffusion problem the aim of Virtual element formulation is to define a discrete space such that for some $k \in \mathbb{N}^+$ $[\mathbb{P}_k(\Omega)]^2 \subset \mathbf{V}_h$ where $\mathbf{V}_h$ represents the virtual element space. In this case, as for Navier-Stokes problem, quadratic elements will be used: this means $k = 2$ which is the lowest possible value to define virtual element space chosen in Beirão da Veiga et al. [2017]. Here it is followed the construction done in Beirão da Veiga et al. [2017] as already declared. Let now be the sequence of decomposition $\{\mathcal{T}_h\}_h$ of domain Ω in polygonal elements K with:

$$h_K := \text{diameter}(K), \quad h := \sup_{K \in \mathcal{T}_h} h_K \tag{3.15}$$

Moreover (x_K, y_K) will indicate the coordinates of the centroid of element K in the whole chapter. Here conditions about elements K are exactly as in the case of diffusion problem. Therefore the geometrical shape of elements is a rectangle with the possibility of hanging nodes (limited by Assumption 2). Also the geometry is the one described in the previous chapter. The following propositions are assumed for every h (and so for every mesh) and for every elements K (see Beirão da Veiga et al. [2017]):

Assumption 3 *1. the element K is star-shaped respect to a ball of radius $\geq \gamma h_K$ for some $\gamma \in \mathbb{R}$*

2. the distance between two vertices of element K is $\geq \eta h_K$

It is important to underline that the domain geometry and the meshes taken into account (with rectangular elements with possible hanging nodes) respect Assumption 3. So it will be considered satisfied.

The following spaces are defined for every element K:

$$\mathbb{B}_k(K) := \{v \in C^0(\partial K) \quad \text{s.t.} \quad v_{|e} \in \mathbb{P}_k(e) \quad \forall\, \text{edge}\, e \subset \partial K\}, \tag{3.16}$$

$$\mathcal{G}_k(K) := \nabla(\mathbb{P}_{k+1}(K)) \subseteq [\mathbb{P}_k(K)]^2 \tag{3.17}$$

$$\mathcal{G}_k(K)^\perp \subseteq [\mathbb{P}_k(K)]^2 \quad \text{the } L^2\text{-orthogonal complement to } \mathcal{G}_k(K). \tag{3.18}$$

In the case $k = 2$ the local virtual element finite dimensional space approximating space $\mathbf{V}$ is:

$$\begin{aligned} \mathbf{V}_h^K := \{\mathbf{v} \in [H^1(K)]^2 \quad \text{s.t} \quad \mathbf{v}_{|\partial K} \in [\mathbb{B}_2(\partial K)]^2, \\ -\nu \boldsymbol{\Delta} \mathbf{v} - \nabla s \in \mathcal{G}_0(K)^\perp \quad \text{for some } s \in L^2(K), \quad \text{div}\mathbf{v} \in \mathbb{P}_1(K).\} \end{aligned} \tag{3.19}$$

The first part says that functions belonging to $\mathbf{V}_h^K$ are continuous function on the boundary of element K and polynomials of maximum degree 2 on every edge. The second part asks the function $\mathbf{v} \in \mathbf{V}_h^K$ to be a solution for a Stokes-like problem. $\mathcal{G}_0(K)^\perp$ space dimension is 0, therefore it is equivalent to write $-\nu \boldsymbol{\Delta} \mathbf{v} - \nabla s = 0$.

Moreover the finite-dimensional space for element K approximating Q (Q^N) locally is defined as:

$$Q_h^K := \mathbb{P}_1(K) \tag{3.20}$$

It is possible to observe that $[\mathbb{P}_2(K)]^2 \subseteq \mathbf{V}_h^K$. Moreover it holds that:

$$\dim([\mathbb{B}_2(\partial K)]^2) = 2n_K 2 \tag{3.21}$$

where n_K is the number of vertices (or edges) of polygon K. Without hanging nodes, one has $n_K = 4$, while in the other case $4 \leq n_K \leq 8$ for Assumption 2. As done in Beirão da Veiga et al. [2017] it is possible to conclude that:

$$\dim \mathbf{V}_h^K = \dim([\mathbb{B}_2(\partial K)]^2) + \dim(\mathbb{P}_1(K)) - 1 = 2n_K 2 + 2, \tag{3.22}$$

while

$$\dim(Q_h^K) = \dim(\mathbb{P}_1(K)) = 3. \tag{3.23}$$

Now it is possible to identify a function $\mathbf{v} \in \mathbf{V}_h^K$ from the following degrees of freedom (Dv). They can be divided in four groups (Beirão da Veiga et al. [2017]), however in this case ($k = 2$), one group does not give any contribution so there exist three types of degrees of freedom.

1. $Dv1$: values of $\mathbf{v}$ in the vertices of polygon K
2. $Dv2$: values of $\mathbf{v}$ in the midpoints of edges of polygon K (in one-to-one correspondence with edges of K)
3. $Dv3$: the moments up to order 1 and greater than zero of the divergence of $\mathbf{v}$ in K:

$$\int_K (\mathrm{div}\mathbf{v})\, q_1 \, dK \quad \forall q_1 \in \mathbb{P}_1(K)/\mathbb{R} \tag{3.24}$$

$\mathbb{P}_1(K)/\mathbb{R}$ means all polynomials of degree ≤ 1 without the constants. For any $q \in Q_h^K$ the degrees of freedom are:

- Dq: the moments up to order 1 of q in K:

$$\int_K q\, p_1 \, dK \quad \forall p_1 \in \mathbb{P}_1(K) \tag{3.25}$$

Degrees of freedom Dq are unisolvent for Q_h^K so they identify any element $q \in Q_h^K$. Furthermore, it holds as proven Beirão da Veiga et al. [2017]:

Proposition 5 *The set of degrees of freedom Dv are unisolvent for the local virtual element space $\mathbf{V}_h^K$.*

It is possible to define the global virtual element spaces as:

$$\mathbf{V}_h := \{\mathbf{v} \in \mathbf{V} \quad \text{s.t.} \quad \mathbf{v}_{|K} \in \mathbf{V}_h^K \quad \forall K \in \mathcal{T}_h\} \tag{3.26}$$

and (with only Dirichlet conditions)

$$Q_h := \{q \in L_0^2(\Omega) \quad \text{s.t.} \quad q_{|K} \in Q_h^K \quad \forall K \in \mathcal{T}_h\} \tag{3.27}$$

or (with Neumann conditions):

$$Q_h^N := \{q \in L^2(\Omega) \quad \text{s.t.} \quad q_{|K} \in Q_h^K \quad \forall K \in \mathcal{T}_h\} \tag{3.28}$$

The global degrees of freedom are obtained extending the local ones. They are, for $\mathbf{v}_h \in \mathbf{V}_h$, the value of $\mathbf{v}_h$ (both components) on internal vertices ("internal" means that they are not on Dirichlet boundary $\partial\Omega_D$), on the midpoints (both components) of each internal edge and for every element two moments of divergence as explained above. In particular for a given element K these two degrees of freedom will be:

$$\int_K \mathrm{div}\mathbf{v}_h \frac{(x - x_K)}{\int_K (x - x_K)^2 \, dK} \, dK \qquad \int_K \mathrm{div}\mathbf{v}_h \frac{(y - y_K)}{\int_K (y - y_K)^2 \, dK} \, dK \tag{3.29}$$

where x and y are cartesian coordinates while x_K and y_K are the two coordinates of the centroid of K. The integrals inside denominators can be explicitly computed:

$$\int_K (x - x_K)^2 \, dK = \frac{2}{3} h (l/2)^3 \qquad \int_K (y - y_K)^2 \, dK = \frac{2}{3} l (h/2)^3 \tag{3.30}$$

where h and l in this case indicate the height and the length of element K. The dimension of the space $\mathbf{V}_h$, coherently with the description of global degrees of freedom, is:

$$\mathrm{dim}\mathbf{V}_h = 2n_P + 2n_V + 2n_E \tag{3.31}$$

where n_V,n_E are the numbers of internal vertices and internal edges respectively, while n_P is the number of elements of $\mathcal{T}_h$. For what concerns space Q_h its dimension is:

$$\mathrm{dim}(Q_h) = 3n_P - 1 \tag{3.32}$$

In fact on each element K $q \in Q_h$ belongs to $\mathbb{P}_1(K)$. The "-1" comes from the fact that $q \in L^2_0$ and so $\int_\Omega q \, d\Omega = 0$, therefore knowing moments $\int_K q \, dK$ $\forall K \in \mathcal{T}_h$ except for one K', thanks to the previous property, it is possible to compute the moment of that element K' as:

$$\int_{K'} q \, dK = - \sum_{K \in \mathcal{T}_h / K'} \int_K q \, dK \tag{3.33}$$

If Neumann condition are present it is considered space Q_h^N whose dimension is:

$$\mathrm{dim}(Q_h^N) = 3n_P \tag{3.34}$$

because there is no addition condition pressure.

An important property holds (it follows from (3.19) and (3.20)):

$$\mathrm{div}(\mathbf{V}_h) \subseteq Q_h \tag{3.35}$$

Canonical basis functions

Indicating with N_{dofs} the number of global degrees of freedom for $\mathbf{V}_h$ and N^K the number of local degrees of freedom on element K for $\mathbf{V}_h$ the global canonical basis $\{\boldsymbol{\varphi}_i\}_{i=1}^{N_{dofs}}$ and the local one $\{\boldsymbol{\varphi}_i\}_{i=1}^{N_K}$:

$$\mathrm{dof}_j \boldsymbol{\varphi}_i = \delta_{ij} \tag{3.36}$$

The same definition is applied for the canonical basis on Q_h (Q_h^N) and Q_h^K, respectively, $\{q_i\}_{i=1}^{Q_{dofs}}$ ($\{q_i\}_{i=1}^{Q_{dofs}+1}$) and $\{q_i\}_{i=1}^{Q_K}$ where Q_{dofs} and $Q_K = 3$ are the numbers of global and local degrees of freedom. It is possible to write explicitly the basis functions from the definition. Locally, on a element K, the three basis functions are:

$$q_1 = 1/|K|, \quad q_2 = \frac{(x - x_K)}{\int_K (x - x_K)^2} \quad q_3 = \frac{(y - y_K)}{\int_K (y - y_K)^2} \tag{3.37}$$

where $|K|$ is the area of element K.

Discrete bilinear forms

As far as the $b(\cdot,\cdot)$ form is concerned, no approximation is done. Therefore for the approximate form it would hold $b_h(\cdot,\cdot) = b(\cdot,\cdot)$. Through degrees of freedom previously defined it is possible to compute:

$$b(\mathbf{v},q) = \sum_{K\in\mathcal{T}_h} b^K(\mathbf{v},q) = \sum_{K\in\mathcal{T}_h} \int_K \mathrm{div}\mathbf{v}\, q\, dK \quad \forall\, \mathbf{v}\in\mathbf{V}_h,\, q\in Q_h(Q_h^N) \tag{3.38}$$

To build the discrete bilinear form $a_h(\cdot,\cdot) : \mathbf{V}_h \times \mathbf{V}_h \to \mathbb{R}$ approximating $a(\cdot,\cdot)$ the procedure is similar to the one done for scalar diffusion problem, the main differences come from the vector-valued Laplacian and by the use of degrees of freedom which involve moments of divergence of a function and not the moments of the function itself. Using degrees of freedom, $a^K(\mathbf{q}_2,\mathbf{v})$ can be computed $\forall\mathbf{q}_2 \in [\mathbb{P}_2(K)]^2$ and $\forall\mathbf{v}\in\mathbf{V}_h^K$.

$$a^K(\mathbf{q}_2,\mathbf{v}) = \nu\left(\int_K \nabla\mathbf{q}_2 : \nabla\mathbf{v}\, dK\right) = \nu\left(-\int_K \boldsymbol{\Delta}\mathbf{q}_2\cdot\mathbf{v}\, dK + \int_{\partial K}(\nabla\mathbf{q}_2\mathbf{n})\cdot\mathbf{v}\, ds\right) \tag{3.39}$$

where ":" indicates the tensorial scalar product. $\boldsymbol{\Delta}\mathbf{q}_2 \in [\mathbb{P}_0(K)]^2$, therefore it is possible to write:

$$\boldsymbol{\Delta}\mathbf{q}_2 = \nabla q_1 \tag{3.40}$$

where $q_1 \in \mathbb{P}_1(K)/\mathbb{R}$ (it is a polynomial of degree 1 and not a constant). Continuing calculation:

$$\begin{gathered} a^K(\mathbf{q}_2,\mathbf{v}) = \nu\left(-\int_K \nabla q_1\cdot\mathbf{v}\, dK + \int_{\partial K}(\nabla\mathbf{q}_2\mathbf{n})\cdot\mathbf{v}\, ds\right) = \\ \nu\left(\int_K q_1\mathrm{div}\mathbf{v}\, dK + \int_{\partial K}(\nabla\mathbf{q}_2\mathbf{n} - q_1\mathbf{n})\cdot\mathbf{v}\, ds\right) \end{gathered} \tag{3.41}$$

The first part is an integral on the whole element area and can be computed using divergence moments ($Dv3$). The other part is an integral on the boundary and it can be computed using $Dv1$ and $Dv2$.

As in the case on diffusion for the discrete form:

$$a_h^K(\cdot,\cdot) : \mathbf{V}_h^K \times \mathbf{V}_h^K \to \mathbb{R} \tag{3.42}$$

it is necessary to introduce an operator because $a_h^K(v_h, w_h)$, with v_h,$w_h \in \mathbf{V}_h^K$, is not computable. The discrete bilinear form must satisfy the properties:

1. *k-consistency* (in this case $k = 2$):

$$a_h^K(\mathbf{q}_2,\mathbf{v}_h) = a^K(\mathbf{q}_2,\mathbf{v}_h) \quad \forall\mathbf{q}_2\in[\mathbb{P}_2(K)]^2, \quad \forall\mathbf{v}_h\in\mathbf{V}_h^K \tag{3.43}$$

2. *stability*: there exist two positive constant α_* and α^* such that:

$$\alpha_* a^K(\mathbf{v}_h,\mathbf{v}_h) \le a_h^K(\mathbf{v}_h,\mathbf{v}_h) \le \alpha^* a^K(\mathbf{v}_h,\mathbf{v}_h) \quad \forall\mathbf{v}_h\in\mathbf{V}_h^K \tag{3.44}$$

Nabla operator Again, as for scalar diffusion, *Nabla operator* on K $\mathbf{\Pi}_2^{\nabla,K} : \mathbf{V}_h^K \to [\mathbb{P}_2(K)]^2$ is defined (see Beirão da Veiga et al. [2017]) given a function $\mathbf{v}_h \in \mathbf{V}_h$:

$$\begin{cases} a^K(\mathbf{q}_2, \mathbf{v}_h) = a^K(\mathbf{q}_2, \mathbf{\Pi}_2^{\nabla,K}\mathbf{v}_h) \quad \forall \mathbf{q}_2 \in [\mathbb{P}_2(K)]^2, \\ \mathbf{P}^{0,K}(\mathbf{v}_h) = \mathbf{P}^{0,K}(\mathbf{\Pi}_2^{\nabla,K}\mathbf{v}_h) \end{cases} \tag{3.45}$$

where $\mathbf{P}^{0,K}$ is the L^2 projector on constants on the element K. Looking at the definition of analogous projector for the scalar case, $\mathbf{P}^{0,K}(\mathbf{v}_h)$ is computed as follows.

$$P_x^{0,K}(\mathbf{v}_h) = \frac{1}{|K|}\int_K [1{,}0] \cdot \mathbf{v}_h, \quad P_y^{0,K}(\mathbf{v}_h)\, dK = \frac{1}{|K|}\int_K [0{,}1] \cdot \mathbf{v}_h\, dK \tag{3.46}$$

where subscripts $(\cdot)_x$ and $(\cdot)_y$ indicate respectively the first and the second component of $\mathbf{P}^{0,K}(\mathbf{v}_h)$. It is possible to observe that $\mathbf{\Pi}_2^{\nabla,K}\mathbf{q}_2 = \mathbf{q}_2 \; \forall \mathbf{q}_2 \in \mathbb{P}_2(K)$.

Therefore if $a_h^K(\mathbf{u}_h, \mathbf{v}_h)$ is defined as $a^K(\mathbf{\Pi}_2^{\nabla,K}\mathbf{u}_h, \mathbf{\Pi}_2^{\nabla,K}\mathbf{v}_h)$ for generic function $\mathbf{u}_h, \mathbf{v}_h \in \mathbf{V}_h^K$, it satisfies *k-consistency* property but it does not respect *stability.* For this reason a stabilizing bilinear form $\mathcal{S}_h^K : \mathbf{V}_h^K \times \mathbf{V}_h^K \to \mathbb{R}$ is added (see Beirão da Veiga et al. [2017]). It must hold:

$$\gamma_* a^K(\mathbf{v}_h, \mathbf{v}_h) \le \mathcal{S}_h^K(\mathbf{v}_h, \mathbf{v}_h) \le \gamma^* a^K(\mathbf{v}_h, \mathbf{v}_h) \quad \forall \, \mathbf{v}_h \in \mathbf{V}_h \quad \text{s.t.} \quad \mathbf{\Pi}_2^{\nabla,K}\mathbf{v}_h = \mathbf{0}. \tag{3.47}$$

where γ_* and γ^* are positive constants independent from h and K. As explained in diffusion chapter, mimicking an identity, the discrete bilinear form $a_h(\cdot, \cdot)$ is defined in the way below:

$$a_h^K(\mathbf{u}_h, \mathbf{v}_h) := a^K(\mathbf{\Pi}_2^{\nabla,K}\mathbf{u}_h, \mathbf{\Pi}_2^{\nabla,K}\mathbf{v}_h) + \mathcal{S}_h^K((\mathbf{I} - \mathbf{\Pi}_2^{\nabla,K})\mathbf{u}_h, (\mathbf{I} - \mathbf{\Pi}_2^{\nabla,K})\mathbf{v}_h) \tag{3.48}$$

In this way the bilinear form $a_h(\cdot, \cdot)$ satisfies *k-consistency* and *stability* properties. As already done for the case of diffusion and using the choice done in Beirão da Veiga et al. [2017] the stabilizing form is defined in the following way:

$$\mathcal{S}_h^K(\mathbf{u}_h, \mathbf{v}_h) = C^K \bar{\mathbf{u}}_h^T \bar{\mathbf{v}}_h \tag{3.49}$$

where $\bar{\mathbf{u}}_h$ and $\bar{\mathbf{v}}_h$ are arrays collecting the degrees of freedom of functions $\mathbf{u}_h, \mathbf{v}_h \in \mathbf{V}_h^K$ and C^K is a positive constant independent from h. C^K is chosen as in Beirão da Veiga et al. [2017]. It is the mean value of the eigenvalues of the matrix coming from $a^K(\mathbf{\Pi}_2^{\nabla,K}\mathbf{u}_h, \mathbf{\Pi}_2^{\nabla,K}\mathbf{v}_h)$.

The global bilinear form $a_h(\cdot, \cdot) : \mathbf{V}_h \times \mathbf{V}_h \to \mathbb{R}$ is set as (see Beirão da Veiga et al. [2017]):

$$a_h(\mathbf{u}_h, \mathbf{v}_h) = \sum_{K \in \mathcal{T}_h} a_h^K(\mathbf{u}_h, \mathbf{v}_h) \quad \forall \mathbf{u}_h, \mathbf{v}_h \in \mathbf{V}_h \tag{3.50}$$

External force approximation

To approximate the right-hand side the local operator (on element K) $\mathbf{\Pi}_0^{0,K} : [L^2(K)]^2 \to [\mathbb{P}_0(K)]^2$ (so the $L^2(K)$ projector on constants) is used. As for the diffusion it is set (see Beirão da Veiga et al. [2017]):

$$\mathbf{f}_h = \mathbf{\Pi}_0^{0,K}\mathbf{f} \quad \forall K \in \mathcal{T}_h \tag{3.51}$$

The right-hand side will be, given a function: $\mathbf{v}_h \in \mathbf{V}_h$ (see Beirão da Veiga et al. [2017]):

$$(\mathbf{f}_h, \mathbf{v}_h) = \sum_{K \in \mathcal{T}_h} \mathbf{f}_h \cdot \mathbf{v}_h \, dK = \sum_{K \in \mathcal{T}_h} \int_K \mathbf{\Pi}_0^{0,K} \mathbf{f} \cdot \mathbf{v}_h \, dK = \sum_{K \in \mathcal{T}_h} \int_K \mathbf{f} \cdot \mathbf{\Pi}_0^{0,K} \mathbf{v}_h \, dK \qquad (3.52)$$

Using the fact that the projection is on constants it is possible to continue the computation:

$$\begin{gathered}(\mathbf{f}_h, \mathbf{v}_h) = \sum_{K \in \mathcal{T}_h} \left((\mathbf{\Pi}_0^{0,K} \mathbf{v}_h)_x \int_K f_x \, dK + (\mathbf{\Pi}_0^{0,K} \mathbf{v}_h)_y \int_K f_y \, dK \right) = \\ \sum_{K \in \mathcal{T}_h} \left(\left(\frac{1}{|K|} \int_K (\mathbf{\Pi}_0^{0,K} \mathbf{v}_h) \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} dK \right) \int_K f_x \, dK + \left(\frac{1}{|K|} \int_K (\mathbf{\Pi}_0^{0,K} \mathbf{v}_h) \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} dK \right) \int_K f_y \, dK \right)\end{gathered} \qquad (3.53)$$

It is possible to compute $\mathbf{\Pi}_0^{0,K} \mathbf{v}_h$ from the degrees of freedom. In particular:

$$\begin{gathered}\int_K \mathbf{\Pi}_0^{0,K} \mathbf{v}_h \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} dK = \int_K \mathbf{v}_h \cdot \nabla (x - x_K) \, dK = \\ - \int_K \operatorname{div} \mathbf{v}_h (x - x_k) \, dK + \int_{\partial K} (x - x_K) \mathbf{v}_h \cdot \mathbf{n} \, ds\end{gathered} \qquad (3.54)$$

The first part is computable using moments of divergence ($Dv3$) while the second one using $Dv1$ and $Dv2$. Analogously:

$$\begin{gathered}\int_K \mathbf{\Pi}_0^{0,K} \mathbf{v}_h \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} dK = \int_K \mathbf{v}_h \cdot \nabla (y - y_K) \, dK = \\ - \int_K \operatorname{div} \mathbf{v}_h (y - y_k) \, dK + \int_{\partial K} (y - y_K) \mathbf{v}_h \cdot \mathbf{n} \, ds\end{gathered} \qquad (3.55)$$

It is now stated a lemma about the error of $\mathbf{f}_h$ (see Beirão da Veiga et al. [2017]):

Lemma 1 *Let $\mathbf{f}_h$ defined in (3.51) assuming $\mathbf{f} \in H^1(\Omega)$. Then, $\forall \, \mathbf{v}_h \in \mathbf{V}_h$, it holds*

$$|(\mathbf{f}_h - \mathbf{f}, \mathbf{v}_h)| \leq C h^2 |\mathbf{f}|_1 ||\mathbf{v}_h||_1 \qquad (3.56)$$

where C is a constant not depending on h.

3.2.2 Discrete Stokes VEM problem

In the end the discrete problem can be written as (see Beirão da Veiga et al. [2017]):

$$\text{find} \quad (\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h(Q_h^N), \quad \text{s.t} \quad \begin{cases} a_h(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = (\mathbf{f}_h, \mathbf{v}_h) & \forall \mathbf{v}_h \in \mathbf{V}_h \\ b(\mathbf{u}_h, q_h) = 0 \quad \forall q_h \in Q_h(Q_h^N) \end{cases} \qquad (3.57)$$

The second equation expresses the incompressibility condition namely the condition about the divergence of velocity. Moreover, together with property (3.35), it ensures that discrete velocity $\mathbf{u}_h$ is *exactly divergence free* (Beirão da Veiga et al. [2017]). To state a theorem about existence and uniqueness of solution it is necessary to discuss *inf-sup condition* done below.

Interpolation result

Using a classical lemma by Brenner-Scott (Brenner and Scott [2008]) in Beirão da Veiga et al. [2017] the following proposition is stated and proven. Under Assumptions 3, that for the used mesh are fulfilled and focusing on $k = 2$ (the polynomial accuracy considered here) it is possible to write:

Proposition 6 *Given $\mathbf{u} \in \mathbf{V} \cap [H^{s+1}(\Omega)]^2$ with $0 \le s \le 2$, there exists $\mathbf{u_I} \in \mathbf{V}_h$ such that*

$$||\mathbf{u} - \mathbf{u_I}||_{0,K} + h_K|\mathbf{u} - \mathbf{u_I}|_{1,K} \le Ch^{s+1}|\mathbf{u}|_{s+1,D(K)} \tag{3.58}$$

where C is a constant independent of h and $D(K)$ is the union of polygons intersecting element K.

Inf-sup condition

The bilinear form $b(\cdot,\cdot) : \mathbf{V}_h \times Q_h \, (Q_h^N) \to \mathbb{R}$ acts on two different spaces, therefore it does not make sense to consider coercivity of the form. However it is possible to prove (see Beirão da Veiga et al. [2017]) that the *inf-sup condition* is valid for the bilinear form.

Proposition 7 *Considering discrete spaces $\mathbf{V}_h$ and $Q_h \, (Q_h^N)$ it is valid that:*

$$\exists \beta > 0 \quad \textit{independent of } h \quad \textit{s.t.} \quad \sup_{\mathbf{v}_h \in \mathbf{V}_h, \mathbf{v}_h \neq \mathbf{0}} \frac{b(\mathbf{v}_h, q_h)}{||\mathbf{v}_h||_1} \ge \beta ||q_h||_Q \quad \forall q_h \in Q_h \, (Q_h^N) \tag{3.59}$$

Existence and uniqueness of discrete problem Thanks to *inf-sup condition* of $b(\cdot,\cdot)$ and considering that $a_h(\cdot,\cdot)$ is stable with respect to $\mathbf{V}$ norm the following theorem about discrete problem (3.57) holds (see Boffi et al. [2013]):

Theorem 4 *It exists a unique solution $(\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h$ of the Stokes VEM discrete problem (3.57). Moreover it holds:*

$$||\mathbf{u}_h||_1 + ||p_h||_Q \le C||\mathbf{f}||_0 \tag{3.60}$$

It is possible to observe that the *inf-sup condition* with property (3.35) implies:

$$\text{div}\mathbf{V}_h = Q_h \tag{3.61}$$

Convergence

Considering Proposition 6 and other results in Beirão da Veiga et al. [2017] two inequalities are reported. Given $\mathbf{v} \in [H^3(\Omega)]^2$ and $q \in H^2(\Omega)$, it is true that:

$$\inf_{\mathbf{v}_h \in \mathbf{V_h}} ||\mathbf{v}_h - \mathbf{v}||_1 \le Ch^2|\mathbf{v}|_3 \qquad \inf_{q_h \in Q_h} ||q_h - q||_Q \le Ch^2|q|_2 \tag{3.62}$$

Let now be defined the spaces (see Beirão da Veiga et al. [2017]):

$$\mathbf{Z} := \{\mathbf{v} \in \mathbf{V} \quad \text{s.t.} \quad b(\mathbf{v}, q) = 0 \quad \forall q \in Q\} \tag{3.63}$$

$$\mathbf{Z}_h := \{\mathbf{v}_h \in \mathbf{V}_h \quad \text{s.t.} \quad b(\mathbf{v}_h, q_h) = 0 \quad \forall q_h \in Q_h \, (Q_h^N)\} \tag{3.64}$$

It is possible to observe that (see Beirão da Veiga et al. [2017]):

$$\mathbf{Z}_h \subseteq \mathbf{Z} \tag{3.65}$$

Now the continuous problem (3.9) can be written as an elliptic problem (see Beirão da Veiga et al. [2017]):

$$\text{find} \quad \mathbf{u} \in \mathbf{Z} \quad \text{s.t.} \quad a(\mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{Z} \tag{3.66}$$

where $\mathbf{u}$ indicates the velocity solution of continuous problem (3.9). Also for discrete problem (3.57) it is possible to formulate an equivalent elliptic problem (see Beirão da Veiga et al. [2017]):

$$\text{find} \quad \mathbf{u}_h \in \mathbf{Z}_h \quad \text{s.t.} \quad a(\mathbf{u}_h, \mathbf{v}_h) = (\mathbf{f}_h, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{Z}_h \tag{3.67}$$

It is possible to say (see Beirão da Veiga et al. [2017]) that $\mathbf{Z}_h$ approximates $\mathbf{Z}$ because (see Boffi et al. [2013]):

$$\inf_{\mathbf{z}_h \in \mathbf{Z}_h} ||\mathbf{z} - \mathbf{z}_h||_1 \leq C \inf_{\mathbf{v}_h \in \mathbf{V}_h} ||\mathbf{z} - \mathbf{v}_h||_1 \tag{3.68}$$

The following theorems describe convergence about velocity and pressure. For observations about their derivation and about the proof of theorem about pressure convergence see Beirão da Veiga et al. [2017]. The theorem about *discrete velocity convergence* is:

Theorem 5 *Let $\mathbf{u} \in \mathbf{Z}$ be the solution to elliptic problem (3.66) which derives from Stokes equations and $\mathbf{u}_h \in \mathbf{Z}_h$ the correspondent discrete solution of problem (3.67). It is true that:*

$$|\mathbf{u} - \mathbf{u}_h|_1 \leq Ch^2(|\mathbf{f}|_1 + |\mathbf{u}|_3) \tag{3.69}$$

Therefore for velocity the error scales as $O(h^2)$ and it does depend on pressure. For what concerns pressure it holds:

Theorem 6 *Let $(\mathbf{u}, p) \in \mathbf{V} \times Q$ be the solution of Stokes continuous problem (3.9) and $(\mathbf{u_h}, p_h) \in \mathbf{V}_h \times Q_h \, (Q_h^N)$ be the solution to discrete problem (3.57). Then:*

$$||p - p_h||_Q \leq Ch^2(|\mathbf{f}|_1 + |\mathbf{u}|_3 + |p|_2) \tag{3.70}$$

Therefore the pressure error scales as $O(h^2)$.

3.3 Construction of the linear Stokes system and details about implementation

3.3.1 Matrix form of the Stokes system

In order to write the discrete problem as a linear system it is necessary to introduce a matrix $\mathbf{A}$ related to discrete bilinear form $a_h(\cdot, \cdot)$ (the global stiffness matrix) and a matrix $\mathbf{P}$ related to discrete bilinear form $b_h(\cdot,\cdot) = b_(\cdot,\cdot)$, which will be called global pressure matrix. Remembering definition of global canonical basis and indicating with $\boldsymbol{\varphi}_i$ the functions of basis of discrete space $\mathbf{V}_h$ and with q_j the functions of basis of discrete space Q_h (Q_h^N) it is possible to define:

$$\mathbf{A}_{ij} = a_h(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i) \quad \text{with} \quad i, j = 1, ..., N_{dofs} \tag{3.71}$$

where N_{dofs} is the total number of global degrees of freedom equal to the dimension of space $\mathbf{V}_h$: $2(n_V + n_E + n_P)$.

Then:

$$\mathbf{P}_{ij} = b(\boldsymbol{\varphi}_i, q_j) \quad \text{with} \quad i = 1, ..., N_{dofs} \quad \text{and} \quad j = 1, ..., Q_{dofs}(+1) \tag{3.72}$$

where Q_{dofs} is the total number of global degrees of freedom for Q_h ("+1" is added for Q_h^N (Neumann conditions)) equal to the dimension of the space : $3(n_P) - 1$.

Considering the discrete velocity solution of problem (3.57) $\mathbf{u}_h$, since $\mathbf{u}_h \in \mathbf{V}_h$ it is possible to expand it through basis:

$$\mathbf{u}_h = \sum_{i=1}^{N_{dofs}} u_i \boldsymbol{\varphi}_i \tag{3.73}$$

where u_i are the components of $\mathbf{u}_h$ with respect to canonical basis. The previous formula is not true if non-homogeneous Dirichlet boundary conditions are used. In that case an additional term $\sum_{j=1}^{N_B} u_j \boldsymbol{\varphi}_j$ should be added where $\boldsymbol{\varphi}_j$ are an extension-like of canonical basis for "false degrees of freedom" defined on vertices and midpoints of edges on Dirichlet boundaries. Indeed u_j are values of $\mathbf{g}$ ($\mathbf{u}$ on $\partial\Omega_D$) on points and midpoints of edges on Dirichlet boundary $\partial\Omega_D$.

Analogously considering the discrete pressure solution of problem (3.57) p_h, since $p_h \in Q_h$:

$$p_h = \sum_{i=1}^{Q_{dofs}} p_i q_i + p_C \tag{3.74}$$

where p_C is:

$$p_C = \frac{1}{|K'|} \int_{K'} p \, dK' \tag{3.75}$$

In other words p_C is the moment of order zero of element K' (here K' is chosen as the last one of elements enumeration). It be obtained from the moments of order zero of other elements of $\mathcal{T}_h$ using the fact that their sum must be zero because $\int_\Omega p \, d\Omega = 0$.

Otherwise, if there are Neumann boundary conditions,

$$p_h = \sum_{i=1}^{Q_{dofs}+1} p_i q_i \tag{3.76}$$

Now restricting to a element K of decomposition $\mathcal{T}_h$ it is possible to define the local matrices $\mathbf{A}^K$ and $\mathbf{P}^K$:

$$\mathbf{A}_{ij}^K = a_h(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i) \quad \text{with} \quad i, j = 1, ..., N^K \tag{3.77}$$

where N^K is the number of local (on element K) degrees of freedom for discrete space $\mathbf{V}_h^K$ equal to $4N_V^K + 2$, where N_V^K is the number of vertices (and so the edges) of polygon K. Given a function $\mathbf{v}_h \in \mathbf{V}_h^K$ the local degrees are ordered in the following way: the first N_V^K ones are the values of component x of $\mathbf{v}_h$ in the vertices, the second N_V^K are the values of component x of $\mathbf{v}_h$ in the medium point of each edge of K, the following $2N_V^K$ are the same as previous ones for component y, finally the last two are related to momentum of divergence as defined in previous section. Coherently the first $2N_V^K$ functions of canonical basis will have the component y equal to 0, while the second $2N_V^K$ ones will have the component x equal to 0

Similarly:

$$\mathbf{P}_{ij}^K = b(\boldsymbol{\varphi}_i, q_j) \quad \text{with} \quad i = 1, ..., N^K \quad \text{and} \quad j = 1, ..., Q^K \tag{3.78}$$

where Q^K is the number of local (on element K) degrees of freedom for discrete space Q_h^K equal to 3. Given a function $q_h \in Q_h^K$ the first local degree of freedom is the moment of order 0, the second one the moment with $(x - x_K)$ and the last one the moment with $(y - y_K)$. Correspondent basis functions are ordered in the same way.

It is important to repeat that the "-1" in the dimension of Q_h is due to the fact that the moment of order zero of a chosen element K' can be determined using the moments of order zero of the other elements as explained previously. Here the element K' chosen is the last one of numeration. The global degrees of freedom follow the order of elements: for each one the related local degrees of freedom are reported except for last element K' where moment of order 0 is not present. If Q_h^N such consideration must not be done because moment of order zero of element K' is actually a real degree of freedom as it happens for all other elements.

During matrices construction it is built a matrix $\tilde{\mathbf{P}}$ which includes the moment of order zero of K'. Moreover also $\mathbf{A}$ is obtained "deactivating" (when there are Dirichlet boundary condition) rows and columns of a matrix $\tilde{\mathbf{A}}$ which includes also the "degrees of freedom" ans so ("basis function") on the boundary. Therefore $\tilde{\mathbf{A}}$ has row and columns related to additional functions $\boldsymbol{\varphi}_j$, corresponding to vertices and medium points of edges on Dirichlet boundary. Furthermore in $\tilde{\mathbf{P}}$ which is the matrix coming from the assembling of local $\mathbf{P}^K$ the "degrees of freedom" on Dirichlet boundary are considered.

Let now be the array $\underline{\mathbf{f}}$ defined as:

$$\underline{\mathbf{f}}_i^K = (\mathbf{f}_h, \boldsymbol{\varphi}_i) \tag{3.79}$$

using again canonical basis of $\mathbf{V}_h^K$.

Now it is indicated with $\underline{\mathbf{u}}$ the array whose entries are given by components of $\mathbf{u}_h$ respect to canonical basis and with $\underline{\tilde{\mathbf{u}}}$ its extension which includes also the values of $\mathbf{u}_h$ on Dirichlet boundary. Similarly $\underline{\mathbf{p}}$ is the array whose entries are given by components of p_h respect to canonical basis and $\underline{\underline{\tilde{\mathbf{p}}}}$ is the extension which includes moment of order zero of p_h on last element K'.

It is possible to write the discrete Stokes system using canonical basis in the following way:

$$\text{find} \quad (\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h(Q_h^N) \quad \text{s.t.}$$
$$\begin{cases} a_h(\mathbf{u}_h, \boldsymbol{\varphi}_i) + b(\boldsymbol{\varphi}_i, p_h) = (\mathbf{f}, \boldsymbol{\varphi}_i) \quad \text{for} \quad i = 1, ..., N_{dofs} \\ b(\mathbf{u}_h, q_j) = \mathbf{0} \quad \text{for} \quad j = 1, ..., Q_{dofs}(+1) \end{cases} \tag{3.80}$$

Looking at (3.80) and indicating with $\bar{\mathbf{A}}$ a matrix equal to $\tilde{\mathbf{A}}$ and with $\bar{\mathbf{P}}$ a matrix equal to $\tilde{\mathbf{P}}$ both of them without rows related to "degrees of freedom" on Dirichlet boundary it is possible to write the following matrix form system when there are Neumann conditions.

$$\begin{cases} \bar{\mathbf{A}}\underline{\tilde{\mathbf{u}}} + \bar{\mathbf{P}}\underline{\tilde{\mathbf{p}}} = \underline{\mathbf{f}} \\ \tilde{\mathbf{P}}^T\underline{\tilde{\mathbf{u}}} = \mathbf{0} \end{cases} \tag{3.81}$$

Separating Dirichlet boundaries contributions from real degrees of freedom:

$$\begin{cases} \mathbf{A}\underline{\mathbf{u}} + \mathbf{P}\underline{\tilde{\mathbf{p}}} = \underline{\mathbf{f}} - \bar{\mathbf{A}}(:, boun)\underline{\tilde{\mathbf{u}}}(boun) \\ \mathbf{P}^T\underline{\mathbf{u}} = -\tilde{\mathbf{P}}^T(:, boun)\underline{\tilde{\mathbf{u}}}(boun) \end{cases} \tag{3.82}$$

where *boun* indicates the "false" degrees of freedom on Dirichlet boundaries. ":" has the same meaning as in *Matlab*, namely it indicates that all rows/columns are considered. Furthermore it is important to observe that in this case (there are Neumann conditions) $\mathbf{P} = \bar{\mathbf{P}}$. Moreover here $\underline{\tilde{\mathbf{p}}}$ is the vector solution of interest for pressure.

If there are only Dirichlet boundary conditions the term $\bar{\mathbf{P}}\underline{\tilde{\mathbf{p}}}$ must be rewritten taking into account condition $\int_\Omega p \, d\Omega = 0$. It is true that:

$$\bar{\mathbf{P}}\underline{\tilde{\mathbf{p}}} = \mathbf{P}\underline{\mathbf{p}} + \bar{\mathbf{P}}(:, K')p'_C \tag{3.83}$$

where K' indicated the column related to moment of order zero of element K' namely the third last column. p'_C is equal to moment of order zero of element K', therefore it can be computed as $-\sum_{m=1}^{N_P-1} p_{m0}$ where p_{m0} are moments of order zero on the first $N_P - 1$ elements (N_P is the number of elements of $\mathcal{T}_h$). Considering $\underline{\mathbf{p}} \in \mathbb{R}^{Q_{dofs}\times 1}$ it is possible to write p_C as:

$$p'_C = \underline{\mathbf{r}}\underline{\mathbf{p}} \tag{3.84}$$

where $\underline{\mathbf{r}} \in \mathbb{R}^{1\times Q_{dofs}}$ defined as:

$$\underline{\mathbf{r}} = \begin{bmatrix} 1 & 0 & 0 & ... & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.85}$$

In other words the vector as non zero entries for the ones corresponding to moments of order zero of pressure. Now formula (3.83) can be written as:

$$\begin{aligned} &\mathbf{P}\underline{\mathbf{p}} + \bar{\mathbf{P}}(:, K')\,\underline{\mathbf{r}}\,\underline{\mathbf{p}} = \\ &\underbrace{(\mathbf{P} + \bar{\mathbf{P}}(:, K')\underline{\mathbf{r}})}_{\mathbf{P}^*}\,\underline{\mathbf{p}} \end{aligned} \tag{3.86}$$

where $\underline{\mathbf{p}}$ appears explicitly in every term.
Therefore the final linear system when there are only Dirichlet conditions is:

$$\begin{cases} \mathbf{A}\underline{\mathbf{u}} + \mathbf{P}^*\underline{\mathbf{p}} = \underline{\mathbf{f}} - \bar{\mathbf{A}}(:, boun)\underline{\tilde{\mathbf{u}}}(boun) \\ \mathbf{P}^T\underline{\mathbf{u}} = -\tilde{\mathbf{P}}^T(:, boun)\underline{\tilde{\mathbf{u}}}(boun) \end{cases} \tag{3.87}$$

An example of how the final system looks like can be found in Beirão da Veiga et al. [2015].

Neumann conditions Is is important to underline that Neumann condition are implemented only for outflow boundary, therefore this is equivalent to impose, considering that normal vector is $\mathbf{n} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$,

$$\nu \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial x} \end{bmatrix} + \begin{bmatrix} p \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.88}$$

Local matrices and right-hand side

Global matrices are obtained starting from local ones. Given an element $K \in \mathcal{T}_h$ let be $\{\boldsymbol{\varphi}_i\}_{i=1,\dots,N^K}$ the local canonical basis functions for space $\mathbf{V}_h^K$ which are always ordered as correspondent degrees of freedom. Furthermore let be q_j the local canonical basis functions for space Q_h^K defined explicitly in (3.37).

Local stiffness matrix $\mathbf{A}^K$ defined in (3.77) is built extending what was done for bilinear form of diffusion case (following the work in Beirão da Veiga et al. [2014]) to vector-valued functions and adapting it to the new degrees of freedom defined for Stokes case. First of all the vector-valued monomial space $\mathcal{M}_2^2(K)$ is chosen as basis of $[\mathbb{P}_2(K)]^2$. $\mathcal{M}_2^2(K)$ definition is similar to the one of space $\mathcal{M}_1^2(K)$ introduced for computing gradient error in diffusion case. In this case non-zero entries of vector-valued functions are polynomial of degree ≤ 2, so space $\mathcal{M}_2^2(K)$ has dimension equal to 12 with the first 6 functions with second component equal to zero.

Considering Nabla operator (3.45) $\boldsymbol{\Pi}_2^{\nabla,K} : \mathbf{V}_h \to [\mathbb{P}_2(K)]^2$ it is clear that $\boldsymbol{\Pi}_2^{\nabla,K}\boldsymbol{\varphi}_i \in [\mathbb{P}_2(K)]^2$, therefore

$$\boldsymbol{\Pi}_2^{\nabla,K}\boldsymbol{\varphi}_i = \sum_{\beta=1}^{12} s_i^\beta \mathbf{m}_\beta \tag{3.89}$$

where m_β are the monomials in $\mathcal{M}_2^2(K)$ and s_i^β are components of $\boldsymbol{\varphi}_i$ related to that basis. From the definition of Nabla operator (3.45) it is possible to write:

$$\sum_{\beta=1}^{12} s_i^\beta a^K(\mathbf{m}_\alpha, \mathbf{m}_\beta) = a^K(\mathbf{m}_\alpha, \boldsymbol{\varphi}_i), \quad \text{for} \quad \alpha = 2, \dots,6,8, \dots 12 \tag{3.90}$$

which is valid for $i = 1, \dots, N^K$. In other words $\mathbf{m}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\mathbf{m}_7 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ are not considered in the previous equation because they would lead to the trivial identity $0 = 0$. Those cases are described by the second equation in (3.45). In particular it is possible to write:

$$\begin{aligned} \sum_{\beta=1}^{12} s_i^\beta P_x^{0,K}(\mathbf{m}_\beta) &= P_x^{0,K}(\boldsymbol{\varphi}_i) \quad \text{for} \quad \alpha = 1 \\ \sum_{\beta=1}^{12} s_i^\beta P_y^{0,K}(\mathbf{m}_\beta) &= P_y^{0,K}(\boldsymbol{\varphi}_i) \quad \text{for} \quad \alpha = 7 \end{aligned} \tag{3.91}$$

valid for $i = 1, ..., N^K$ and where $P_x^{0,K}(\cdot)$ and $P_y^{0,K}(\cdot)$ are defined in (3.46). As for diffusion, starting from (3.91) and (3.90) it is possible to write the system

$$\mathbf{G}\mathbf{s}_i = \mathbf{b}^i \quad \text{for} \quad i = 1, ..., N^K \tag{3.92}$$

where $\mathbf{s}_i^\beta = s_i^\beta$ and $\mathbf{G} \in \mathbb{R}^{12\times 12}$ is the matrix such that:

$$\begin{cases} \mathbf{G}_{ij} = a^K(\mathbf{m}_i, \mathbf{m}_j) \quad \text{for} \quad i \neq 1,7 \\ \mathbf{G}_{1j} = P_x^{0,K}(\mathbf{m}_j) \\ \mathbf{G}_{7j} = P_y^{0,K}(\mathbf{m}_j) \end{cases} \tag{3.93}$$

Then, given i ,$1 \leq i \leq N^K$, $\mathbf{b}^i$ is a vector such that:

$$\begin{cases} \mathbf{b}_j^i = a^K(\mathbf{m}_j, \boldsymbol{\varphi}_i) \quad \text{for} \quad j \neq 1,7 \\ \mathbf{b}_1^i = P_x^{0,K}\boldsymbol{\varphi}_i \\ \mathbf{b}_7^i = P_y^{0,K}\boldsymbol{\varphi}_i \end{cases} \tag{3.94}$$

As for diffusion, to write equations in a compact form with all i let be:

$$\mathbf{B} := \left[\mathbf{b}^1, ..., \mathbf{b}^{N^K}\right] \tag{3.95}$$

Therefore

$$\begin{cases} \mathbf{B}_{ij} = a^K(\mathbf{m}_i, \boldsymbol{\varphi}_j) \quad \text{for} \quad i \neq 1,7 \\ \mathbf{B}_{1j} = P_x^{0,K}\left(\boldsymbol{\varphi}_j\right) \\ \mathbf{B}_{7j} = P_y^{0,K}\left(\boldsymbol{\varphi}_j\right) \end{cases} \tag{3.96}$$

So $\mathbf{B} \in \mathbb{R}^{12\times N^K}$.

Now it is possible to build the matrix representation $\mathbf{\Pi}_*^\nabla \in \mathbb{R}^{N^K\times 12}$ of Nabla operator $\mathbf{\Pi}_2^{\nabla,K}$ acting from $\mathbf{V}_h^K$ to $[\mathbb{P}_2(K)]^2$ (described using polynomial basis $\mathcal{M}_2^2(K)$). Therefore $(\mathbf{\Pi}_*^\nabla)_{\beta i} = s_i^\beta$ and it can be obtained, extending what was reported for diffusion case, from:

$$\mathbf{\Pi}_*^\nabla = \mathbf{G}^{-1}\mathbf{B} \tag{3.97}$$

Since $[\mathbb{P}_2(K)]^2 \subset \mathbf{V}_h^K$, $\mathbf{\Pi}_2^{\nabla,K}$ can be seen as a operator acting from $\mathbf{V}_h^K$ to $\mathbf{V}_h^K$ (described using the usual local canonical basis). So, given an element of local canonical basis $\boldsymbol{\varphi}_i$:

$$\Pi_2^{\nabla,K}\boldsymbol{\varphi}_i = \sum_{j=1}^{N^K} t_i^j \boldsymbol{\varphi}_j, \quad \text{for} \quad i = 1, ..., N^K \tag{3.98}$$

where:

$$t_i^j = \text{dof}_j(\mathbf{\Pi}_2^{\nabla,K}\boldsymbol{\varphi}_i) \tag{3.99}$$

As for diffusion let now be the matrix $\mathbf{D} \in \mathbb{R}^{N^K\times 12}$ defined as:

$$\mathbf{D}_{i\alpha} = \text{dof}_i(\mathbf{m}_\alpha) \quad \text{for} \quad i = 1, ..., N^K \tag{3.100}$$

where dof_i is i-th local degree of freedom while $\mathbf{m}_\alpha \in \mathcal{M}_2^2(K)$ is an element of polynomial basis as defined before.

Now it is possible to write, generalizing the formula for diffusion derived in Beirão da Veiga et al. [2014], matrix representation $\mathbf{\Pi}^{\nabla,K}$ of operator $\mathbf{\Pi}_2^{\nabla,K}$ (acting from $\mathbf{V}_h^K$ to $\mathbf{V}_h^K$) as:

$$\mathbf{\Pi}^{\nabla,K} = \mathbf{D}\mathbf{\Pi}_*{}^{\nabla,K} = \mathbf{D}\mathbf{G}^{-1}\mathbf{B} \tag{3.101}$$

The matrix $\mathbf{G}$ can be also calculated using $\mathbf{B}$ and $\mathbf{D}$ generalizing the result for diffusion observed in Beirão da Veiga et al. [2014]:

$$\mathbf{G} = \mathbf{B}\mathbf{D} \tag{3.102}$$

Again also in this case as observed in Beirão da Veiga et al. [2014] matrices $\mathbf{B}$,$\mathbf{D}$ and $\mathbf{G}$ can be computed independently and then their computation can be checked using (3.102). Moreover matrix $\mathbf{G}$ depend only on monomial basis $\mathcal{M}_2^2(K)$ and, from the definition of vector-valued functions in $\mathcal{M}_2^2(K)$, it is:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}^d & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^d \end{bmatrix} \tag{3.103}$$

where $\mathbf{G}^d$ is the "$\mathbf{G}\quad matrix$" computed for diffusion (scalar) case.

Local pressure matrix $\mathbf{P}^K \in \mathbb{R}^{N^K \times 3}$ is defined in (3.78). Therefore first column of $\mathbf{P}^K$ is related to (see (3.37)) q_1 (corresponding to moment of order zero Dq) and second and third columns are related to q_2 and q_3 (3.37). Focusing on the last two rows of $\mathbf{P}^K$ which can be computed using $Dv3$, it is possible to observe that, from definition of $Dv3$ and of canonical basis functions, the only non-zero entries are:

$$\mathbf{P}^K(4N^V + 1{,}2) = 1 \qquad \mathbf{P}^K(4N^K + 2{,}3) = 1 \tag{3.104}$$

For rows from 1 to $4N^V$ entries in columns 2 and 3 are zero thanks to definition of $Dv1$ and $Dv2$. Values $\mathbf{P}_{i1}^K$ in column 1 are obtained solving the integral:

$$\frac{1}{|K|}\int_K \text{div}\boldsymbol{\varphi}_i \, dK \tag{3.105}$$

which can be rewritten using divergence theorem:

$$\frac{1}{|K|}\int_{\partial K} \boldsymbol{\varphi}_i \cdot \mathbf{n} \, ds \tag{3.106}$$

This last integral can be obtained thanks to $Dv1$ and $Dv2$. Again Simpson rule is used for computation and, in this case, it returns the exact value. In fact $\boldsymbol{\varphi}_i \in \mathbf{V}_h^K$ therefore φ_i is a polynomial of degree ≤ 2 on every edge.

Finally the local right-hand side vector defined in (3.79) that can be obtained following explanation in Section 3.2.1 is computed. Integrals which appear in calculation are approximated with a bidimensional Simpson quadrature rule.

3.3.2 Errors

H^1 error (gradient error) of velocity $\mathbf{u}_h$ A computable form of gradient error for velocity is (see Beirão da Veiga et al. [2017]):

$$\mathrm{err}(\mathbf{u}) := \left(\sum_{K \in \mathcal{T}_h} ||\nabla \mathbf{u} - \mathbf{\Pi}_1^{0,K}(\nabla \mathbf{u}_h)||^2_{0,K} \right)^{1/2} \tag{3.107}$$

in fact $\mathbf{u}_h$ is "not explicitly known" (Beirão da Veiga et al. [2017]) inside the domain. $\mathbf{\Pi}_1^{0,K} : [L^2(K)]^{2\times 2} \to [\mathbb{P}_1(K)]^{2\times 2}$ is the tensor-valued L^2-projector operator defined by

$$\int_K \left(\mathbf{T} - \mathbf{\Pi}_1^{0,K}\mathbf{T}\right) : \mathbf{P}_1 \, dK = 0 \quad \forall\, \mathbf{T} \in [L^2(\Omega)]^{2\times 2} \quad \text{and} \quad \forall\, \mathbf{P}_1 \in [\mathbb{P}_1(K)]^{2\times 2} \tag{3.108}$$

This operator is computable from degrees of freedom of space $\mathbf{V}_h^K$. For space $[\mathbb{P}_1(K)]^{2\times 2}$ the following monomial basis $\mathcal{M}_1^{2\times 2}(K)$ is defined as:

$$\begin{aligned} \mathcal{M}_1^{2\times 2}(K) := \{ &\begin{bmatrix} q_i & 0 \\ 0 & 0 \end{bmatrix} \quad \text{with} \quad i = 1{,}2{,}3; \quad \begin{bmatrix} 0 & q_j \\ 0 & 0 \end{bmatrix} \quad \text{with} \quad j = 1{,}2{,}3 \\ &\begin{bmatrix} 0 & 0 \\ q_w & 0 \end{bmatrix} \quad \text{with} \quad w = 1{,}2{,}3 \quad \begin{bmatrix} 0 & 0 \\ 0 & q_z \end{bmatrix} \quad \text{with} \quad z = 1{,}2{,}3 \\ &\qquad\qquad\qquad\qquad \text{where} \quad q_i, q_j, q_w, q_z \in \mathcal{M}_1(K) \} \end{aligned} \tag{3.109}$$

Now it is possible to write:

$$\mathbf{\Pi}_1^{0,K}(\nabla \mathbf{u}_h) = \sum_{i=1}^{12} t^i \mathbf{Q}_i \quad \text{where} \quad \mathbf{Q}_i \in \mathcal{M}_1^{2\times 2}(K) \tag{3.110}$$

Moreover definition (3.108) can be rewritten with $\mathbf{T} = \nabla \mathbf{u}_h$ and with elements of $\mathcal{M}_1^{2\times 2}(K)$:

$$\sum_{i=1}^{12} t^i \int_K \mathbf{Q}_i : \mathbf{Q}_z \, dK = \int_K (\nabla \mathbf{u}_h) : \mathbf{Q}_z \, dK \quad \forall \mathbf{Q}_z \in \mathcal{M}_1^{2\times 2}(K) \tag{3.111}$$

The second term can be computed from degrees of freedom using integration by parts similarly to what is done for $a^K(\cdot,\cdot)$ in Beirão da Veiga et al. [2017]:

$$\int_K (\nabla \mathbf{u}_h) : \mathbf{Q}_z \, dK = -\int_K \mathrm{div}(\mathbf{Q}_z) \cdot \mathbf{u}_h + \int_{\partial K} (\mathbf{Q}_z \mathbf{n}) \cdot \mathbf{u}_h \, ds \tag{3.112}$$

The divergence for a tensor-valued function $\mathbf{T}$:

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \tag{3.113}$$

is defined as

$$\mathrm{div}\mathbf{T} = \begin{bmatrix} \frac{\partial T_{11}}{\partial x} + \frac{\partial T_{12}}{\partial y} \\ \frac{\partial T_{21}}{\partial x} + \frac{\partial T_{22}}{\partial y} \end{bmatrix} \tag{3.114}$$

$\mathrm{div}(\mathbf{Q}_z)$, due to how $\mathbf{Q}_z$ are defined, is a vector-valued constant function (or $\mathbf{0}$) with only one component different from zero. Therefore if $\mathrm{div}(\mathbf{Q}_z) \neq \mathbf{0}$ it is possible to write:

$$\mathrm{div}(\mathbf{Q}_z) = \nabla q_1 \tag{3.115}$$

where q_1 is a polynomial of degree 1. Resuming calculation:

$$\begin{aligned}\int_K (\nabla \mathbf{u}_h) : \mathbf{Q}_z \, dK = -\int_K \nabla q_1 \cdot \mathbf{u}_h + \int_{\partial K} (\mathbf{Q}_z \mathbf{n}) \cdot \mathbf{u}_h \, ds = \\ \int_K q_1 \mathrm{div}(\mathbf{u}_h) + \int_{\partial K} ((\mathbf{Q}_z \mathbf{n}) - q_1 \mathbf{n}) \cdot \mathbf{u}_h \, ds\end{aligned} \tag{3.116}$$

The first integral is computable from $Dv3$ of $\mathbf{u}_h$ and the second one from $Dv1$ and $Dv2$.

Now let be the matrix $\underline{\mathbf{M}} \in \mathbb{R}^{12\times 12}$ and the vector $\underline{\mathbf{l}} \in \mathbb{R}^{12\times 1}$ be defined as:

$$\underline{\mathbf{M}}_{zi} = \int_K \mathbf{Q}_i : \mathbf{Q}_z \, dK \qquad \underline{\mathbf{l}}_z = \int_K (\nabla \mathbf{u}_h) : \mathbf{Q}_z \, dK \tag{3.117}$$

So (3.111) can be rewritten as a linear system.

$$\underline{\mathbf{M}}\mathbf{t} = \underline{\mathbf{l}} \tag{3.118}$$

where vector $\mathbf{t}$ is defined such that $\mathbf{t}_i = t^i$ and it contains components of $\mathbf{\Pi}_1^{0,K}(\nabla \mathbf{u}_h)$ with respect to basis $\mathcal{M}_1^{2\times 2}(K)$. All integrals on the border, defined above, are computed using Simpson rule.

Now (3.111) can be obtain calculating integral:

$$\int_K \left(\nabla \mathbf{u} - \mathbf{\Pi}_1^{0,K}(\nabla \mathbf{u}_h)\right)^2 \tag{3.119}$$

with a bidimensional Simpson quadature formula.

Pressure (L^2) error L^2 error for pressure scalar field is:

$$\mathrm{err}(p) := ||p - p_h||_0 \tag{3.120}$$

splitting the norm on every element $K \in \mathcal{T}_h$ it is possible to write:

$$\mathrm{err}(p) = \left(\sum_{K \in \mathcal{T}_h} (||p - p_h||_{0,K})^2\right)^{1/2} \tag{3.121}$$

where p_h is a polynomial of degree 1 on K. The integral:

$$\int_K (p - p_h)^2 \, dK \tag{3.122}$$

is approximated using a bidimensional Simpson quadature formula.

3.3.3 Plots

Besides plots already described for diffusion cases about error and functions, here, in order to visualize a velocity field, the *quiver* function (MathWorks) of *Matlab* was utilized showing arrows related to vector field for every node of the mesh. Moreover in order to see streamlines of velocity field in the plot a suitable function was written using *streamline* function (MathWorks) of *Matlab*.

3.4 A posteriori error analysis and adaptive mesh refinement

Adaptive mesh refinement for Stokes case uses the same procedure (**solve-estimate-mark-refine**) as diffusion case with a proper a posteriori error estimate. Moreover the algorithm to refine a mesh element *refine_el* is the same as described in second chapter. The new part with respect to diffusion case is the a posteriori error formulation for Stokes problem which is implemented following theoretical recent work in Wang et al. [2020].

3.4.1 A posteriori error analysis for Stokes problem

The a posteriori error formulation comes from Wang et al. [2020], done with $\nu = 1$, applied to the VEM space used here. In this case estimate is written for general ν. This can be done thinking to Stokes equation in implicit form (all terms are at left-hand side and right-hand side is equal to zero). Then all the equation is divided by ν. In this way the following replacement should be done: $\mathbf{f} \to \mathbf{f}/\nu$, $\mathbf{f}_h \to \mathbf{f}_h/\nu$ and pressure $p_h \to p_h/\nu$.

Given an element $K \in \mathcal{T}_h$ and solutions of discrete VEM problem (3.57) $\mathbf{u}_h$ (velocity) and p_h (pressure) let the error estimator P_K be defined as:

$$P_K^2 = \Theta_K^2 + \eta_{S,K}^2 + \eta_{r,K}^2 \tag{3.123}$$

where Θ_K is the estimate coming from the data of the problem, namely $\mathbf{f}$:

$$\Theta_K^2 = h_K^2 ||\mathbf{f}/\nu - \mathbf{f}_h/\nu||_K^2 \tag{3.124}$$

It is possible to observe that the previous definition is the same as for diffusion case.

Then there is the part of error estimate coming from stabilization term:

$$\eta_{S,K}^2 = \mathcal{S}_h^K((\mathbf{I} - \mathbf{\Pi}_2^{\nabla,K})\mathbf{u}_h, (\mathbf{I} - \mathbf{\Pi}_2^{\nabla,K})\mathbf{u}_h) \tag{3.125}$$

The previous estimate can be computed because operator $\Pi_2^{\nabla,K}$ has already been built in the construction of $a_h(\cdot,\cdot)$ as well as $\mathcal{S}_h^K$.

The error on the element K is given by two parts: one coming from the residual on the element and the other by the "jumps" on internal edges.

$$\eta_{r,K}^2 = h_K^2 ||\mathbf{f}_h/\nu + \Delta\mathbf{\Pi}_2^{\nabla,K}\mathbf{u}_h + \nabla p_h/\nu||_K^2 + \sum_{e \in \partial K \cap E^i} h_e [\![(\nabla\mathbf{\Pi}_2^{\nabla,K}\mathbf{u}_h + (p_h/\nu)\mathbf{I})\mathbf{n}_e]\!]_e^2 \tag{3.126}$$

Again the previous formula can be computed in fact the gradient of p_h can be easily computed because p_h is a polynomial of degree ≤ 1 whose expression is known respect to a basis. E^i denotes the set of internal edges. $[\![\cdot]\!]_e$ is the "jump" on edge e. Let K^+ and K^- be the two elements having edge e in common with, respectively, $\mathbf{n}^+(\mathbf{n}^-)$ as unit outward normal vector on e. Given a function $\mathbf{v}$, indicating with $\mathbf{v}^+(\mathbf{v}^-)$, the restriction of $\mathbf{v}$, respectively, on $K^+(K^-)$, $[\![\mathbf{v}]\!]_e = \mathbf{v}^+\mathbf{n}^+ + \mathbf{v}^-\mathbf{n}^-$. In the computation it is not important which element has positive normal because everything is squared.

Therefore the global a posteriori error P is:

$$P^2 = \sum_{K\in\mathcal{T}_h} P_K^2 \tag{3.127}$$

Now it is possible to state an **upper bound** theorem for P (see Wang et al. [2020]).

Theorem 7 *Let $(\mathbf{u}, p) \in \mathbf{V} \times Q\,(Q^N)$ be the solution of Stokes continuous problem (3.9) and $(\mathbf{u}_h, p_h) \in \mathbf{V} \times Q_h\,(Q_h^N)$ the solution of Stokes discrete problem (3.57). Defined the a posteriori error P as in (3.127) it holds:*

$$|\mathbf{u} - \mathbf{u}_h|_1^2 + ||p - p_h||^2 \lesssim P^2 \tag{3.128}$$

The proof of the theorem can be found in Wang et al. [2020].

3.5 Numerical tests and experiments

As done for the case of diffusion some mathematical cases are considered to validate the used numerical method. Both uniform mesh refinement, where every element is refined, and adaptive mesh refinement, where a posteriori estimator is adopted to refine the mesh, are considered. Moreover in both cases fluid dynamic problem described in introduction is taken into account analyzing the solution and the meshes which come from adaptive mesh refinement.

Plots of velocity field are shown. Furthermore trends of gradient error of velocity (3.107) and L^2-error of pressure (3.120) are reported as well as total error (called in legend *total error*):

$$\textit{total error} = \left(|\mathbf{u} - \mathbf{u}_h|_1^2 + ||p - p_h||_0^2\right)^{1/2} \tag{3.129}$$

In the same plot also a posteriori estimator P defined in (3.127) is shown. As done for diffusion in another plot various contributions of a posteriori estimator are reported separately. Therefore the following definitions are considered.

Θ means:

$$\Theta = \left(\sum_{K\in\mathcal{T}_h} \Theta_K^2\right)^{1/2} \tag{3.130}$$

η_no_edges means:

$$\eta\_no\_edges = \left(\sum_{K\in\mathcal{T}_h} h_K^2 ||\mathbf{f}_h/\nu + \Delta\mathbf{\Pi}_2^{\nabla,K}\mathbf{u}_h + \nabla p_h/\nu||_K^2\right)^{1/2} \tag{3.131}$$

h	gradient error velocity	L^2 error pressure	total error	a posteriori estimator
1/2	4.132e-15	1.728	1.728	6.919
1/4	1.415e-14	0.4327	0.4327	1.731
1/8	6.071e-14	0.1082	0.1082	0.4328

Table 3.1: Gradient errors and a posteriori estimators for uniform refinement sequence: *case 1* (Stokes problem)

η_edges means:

$$\eta\_edges = \left(\sum_{K \in \mathcal{T}_h} \sum_{e \in \partial K \cap E^i} h_e [\![(\nabla \mathbf{\Pi}_2^{\nabla,K} \mathbf{u}_h + (p_h/\nu)\mathbf{I})\mathbf{n}_e]\!]_e^2 \right)^{1/2} \tag{3.132}$$

η_stab means:

$$\eta\_stab = \left(\sum_{K \in \mathcal{T}_h} \eta_{S,K}^2 \right)^{1/2} \tag{3.133}$$

3.5.1 Uniform mesh refinements

For every case 3 meshes are considered with squared elements starting from characteristic dimension $h = 1/2$. It is important to underline that with uniform refinement there are no hanging nodes.

Case 1

In the first case considered the starting mesh is shown in Figure 3.1. $\mathbf{f}$ is such that solution of problem is:

$$\mathbf{u} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad p = y^3 \tag{3.134}$$

$\nu = 1$. Moreover homogeneous Dirichlet boundary conditions are imposed on every boundary. It is possible to observe that, with the domain used:

$$\int_\Omega p \, d\Omega = 0 \tag{3.135}$$

coherently with theoretical formulation for Dirichlet-only boundary conditions. Figure 3.2a and Table 3.1 show the trends of the errors and estimators. Moreover in 3.2b various contributions of a posteriori estimator are distinguished. It is possible to see that only pressure error is important while error on velocity is irrelevant, in fact velocity solution is $[0; 0]$. Pressure error and a posteriori estimator follow slope of line related to $O(h^2)$ convergence. In a posteriori estimator only Θ gives a meaningful contribution.

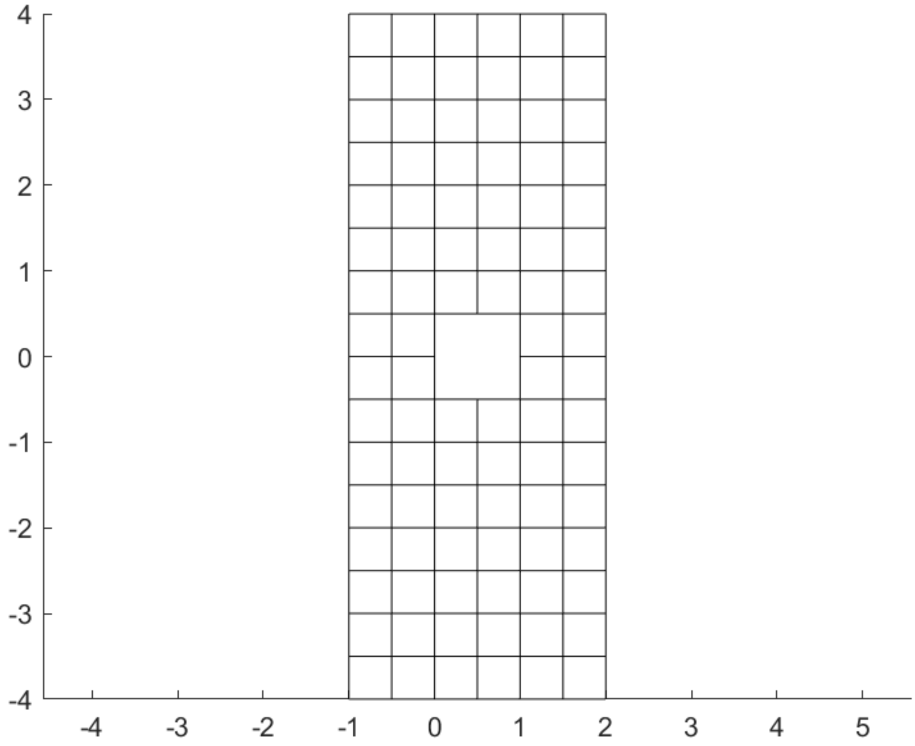
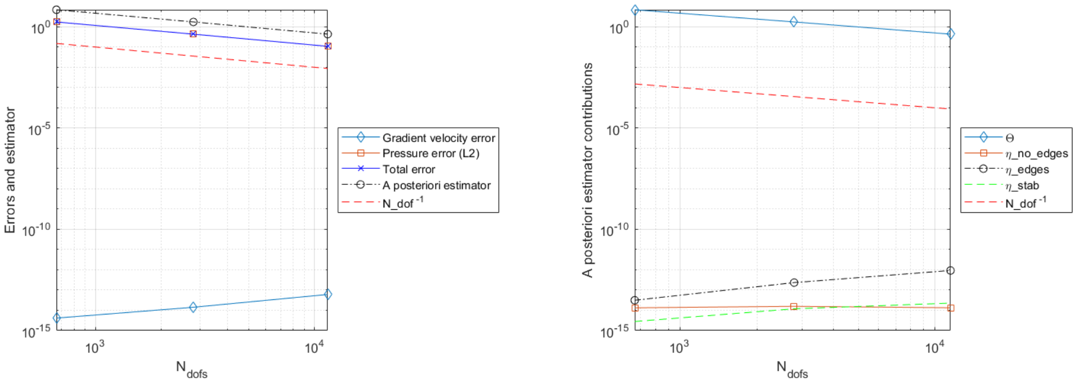


Figure 3.1: Starting mesh used in some of Stokes cases. It is the coarsest mesh with $A = 1$ and $L = 2$



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

Figure 3.2: *Case 1* errors and estimators (Stokes problem)

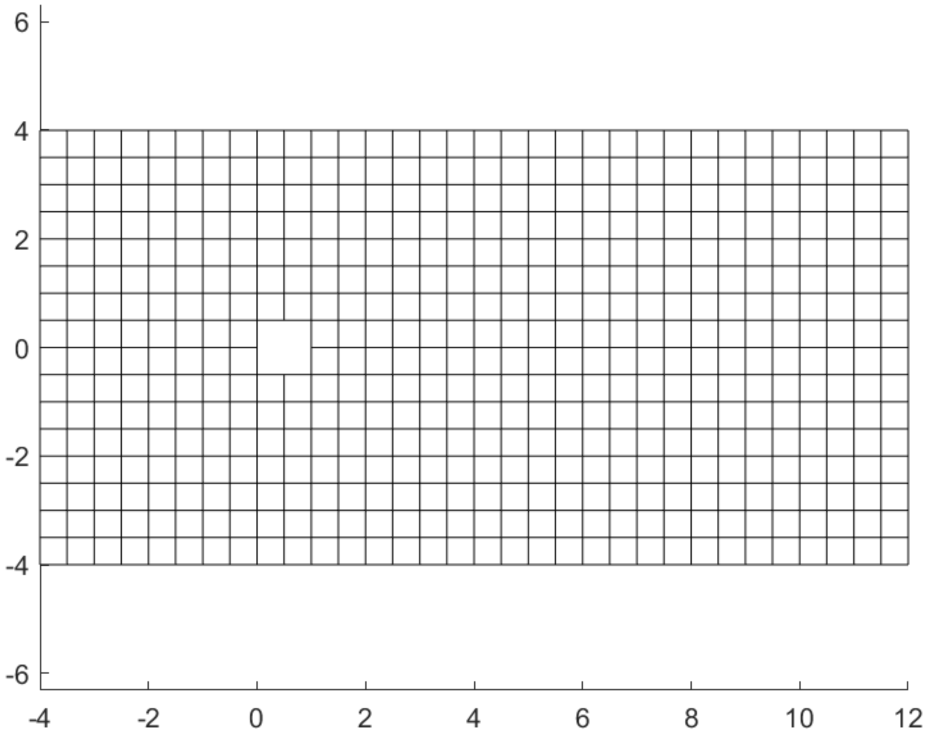


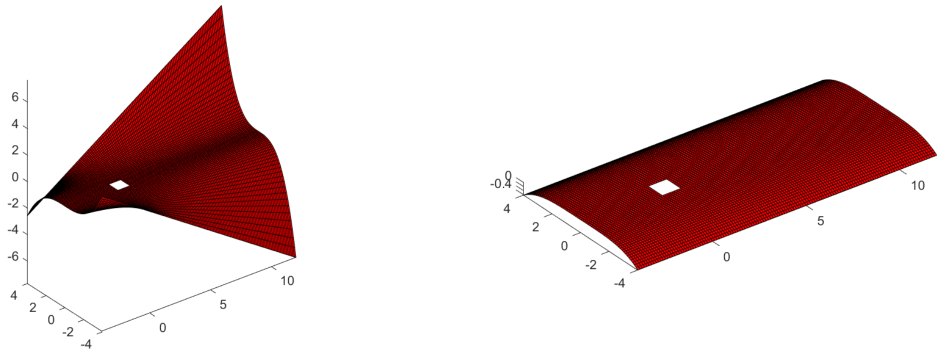
Figure 3.3: Starting mesh used in some of Stokes cases. It is the coarsest mesh with $A = 4$ and $L = 12$

Case 2

In this case the starting mesh considered has $A = 4$ and $L = 12$ (see Figure 3.3). The problem is set such that solution is:

$$\mathbf{u} = \frac{1}{100}\begin{bmatrix} y^3x \\ -y^4/4 \end{bmatrix} \qquad p = -\frac{1}{100}y^3 \tag{3.136}$$

Velocity fulfills the incompressibilty condition div$\mathbf{u} = 0$ and pressure constrain (3.135) is valid. $\nu = 1$. On every boundary non-homogeneous boundary conditions equal to velocity solution are imposed. Figure 3.4 shows plots of the two components of velocity realized with the most refined mesh. Figure 3.5a and Table 3.2 show the trends of the errors and estimators. Moreover in 3.5b various contributions of a posteriori estimator are distinguished. It is possible to see that both velocity and pressure errors follow theoretical results. Moreover the a posteriori estimator is parallel and above the true error as expected. Θ gives the main contribution to the a posteriori estimator which scales as h^2.



(a) u_x

(b) u_y

Figure 3.4: Plots of VEM discrete velocity solution components for case *2* (Stokes problem)

h	gradient error velocity	L^2 error pressure	total error	a posteriori estimator
1/2	0.0765	0.0691	0.103	0.5736
1/4	0.0191	0.0173	0.02586	0.144
1/8	0.00479	0.00433	0.006456	0.03605

Table 3.2: Gradient errors and a posteriori estimators for uniform refinement sequence: *case 2* (Stokes problem)

Case 3

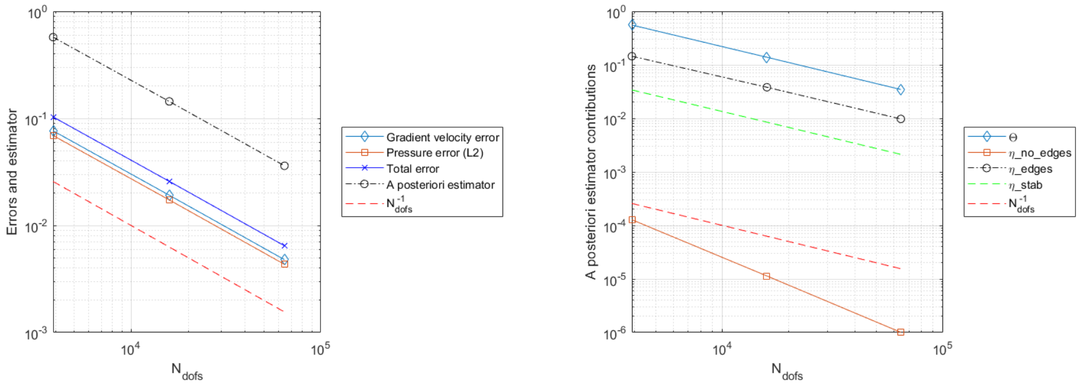
This case considers the same solution (3.136) of *Case 2* but using homogeneous Neumann conditions on outflow boundary. In fact:

$$\begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{1}{100} y^3 \\ 0 \end{bmatrix} = \begin{bmatrix} -p \\ 0 \end{bmatrix} = -p\mathbf{I} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.137}$$

$\mathbf{f}$ does not change from *case 2*. Viscosity is again $\nu = 1$. Plots of solution is not reported because it is very similar to plots of *case 2* (see Figure 3.4). In Figure 3.6a and Table 3.3 errors and a posteriori estimators are reported. A posteriori estimator contributions are shown in Figure 3.6b. The situation is very similar to Dirichlet corresponding case and it allows to validate Neumann conditions on outflow boundary. Errors and estimators again follow theoretical results.

Case 4

Case 4 considers again the same solution as *Case 2* (3.136). Here constant viscosity ν is $\nu = 10$. So, $\mathbf{f}$ is changed coherently with that condition. In other words $\mathbf{f}$ is ten times the function used in *Case 2* and *Case 3*. Non-homogeneous boundary conditions with



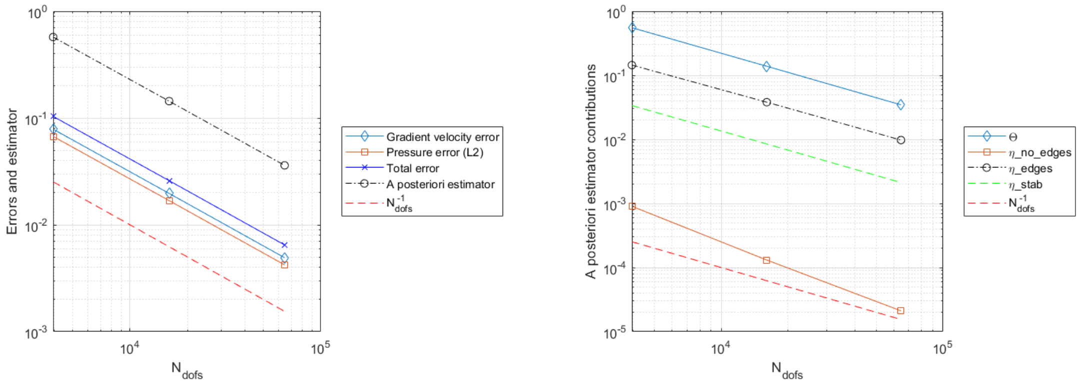
(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

Figure 3.5: *Case 2* errors and estimators (Stokes problem)

h	gradient error velocity	L^2 error pressure	total error	a posteriori estimator
1/2	0.07869	0.0673	0.1036	0.5736
1/4	0.01968	0.0169	0.02592	0.144
1/8	0.00492	0.00422	0.00648	0.03605

Table 3.3: Gradient errors and a posteriori estimators for uniform refinement sequence: *case 3* (Stokes problem)



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

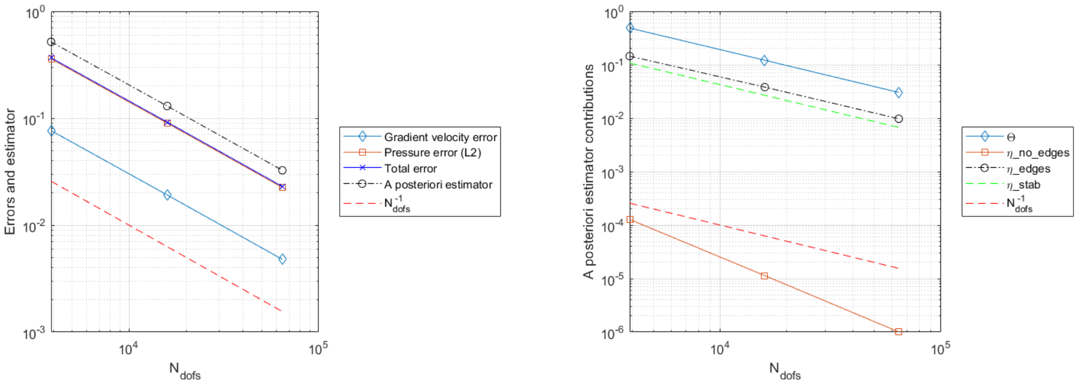
Figure 3.6: *Case 3* errors and estimators (Stokes problem)

velocity solution are used on every boundary. Again plot of the solution is not reported because it is very similar to *Case 2* and *Case 3*.

Figure 3.7a and Table 3.4 report errors and a posteriori estimators. A posteriori estimator contributions are shown in Figure 3.7b. Numerical results are similar to

h	gradient error velocity	L^2 error pressure	total error	a posteriori estimator
1/2	0.07654	0.3589	0.367	0.5178
1/4	0.01914	0.0899	0.0919	0.130
1/8	0.00479	0.02249	0.0230	0.0326

Table 3.4: Gradient errors and a posteriori estimators for uniform refinement sequence: *case 4* (Stokes problem)



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

Figure 3.7: *Case 4* errors and estimators (Stokes problem)

previous cases. Errors follow again theoretical results.

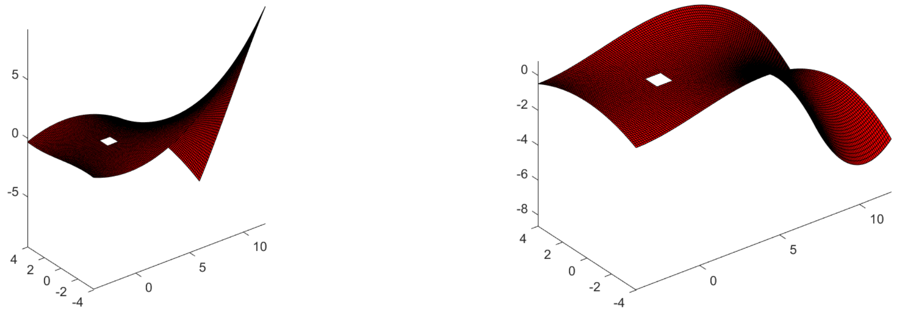
Case 5

Case 5 describes a situation where problem solution is:

$$\mathbf{u} = \frac{1}{100}\begin{bmatrix} \frac{y^3}{2} - \frac{3}{2}x^2y - 2xy \\ -\frac{x^3}{2} + \frac{3}{2}y^2x + y^2 \end{bmatrix} \qquad p = -\frac{1}{100}2y \tag{3.138}$$

In this way $\mathbf{f} = \mathbf{0}$. $\nu = 1$. On boundaries non-homogeneous Dirichlet boundary conditions equal to solution function are imposed. Pressure fulfills (3.135). The starting mesh is shown in Figure 3.3. Figure 3.8 shows plots of components of velocity solution of VEM problem obtained with the most refined mesh. Figure 3.9a and Table 3.5 report errors and a posteriori estimators. A posteriori estimator contributions are shown in Figure 3.9b.

In this case gradient error of velocity and a posteriori estimator follow a usual trend. Error on pressure is negligible in fact pressure solution of continuous problem is a polynomial of degree 1 which belongs to VEM space for pressures Q_h. The main contribution for a posteriori estimator comes from *η_edges*. Θ is zero here because $\mathbf{f} = 0$.



(a) u_x

(b) u_y

Figure 3.8: Plots of VEM discrete velocity solution components for case *5* (Stokes problem)

h	gradient error velocity	L^2 error pressure	total error	a posteriori estimator
1/2	0.01409	1.3073e-13	0.01409	0.04865
1/4	0.003522	1.8464e-13	0.003522	0.01232
1/8	0.0008804	3.998e-12	0.0008804	0.00301

Table 3.5: Gradient errors and a posteriori estimators for uniform refinement sequence: *case 5* (Stokes problem)

Case 6

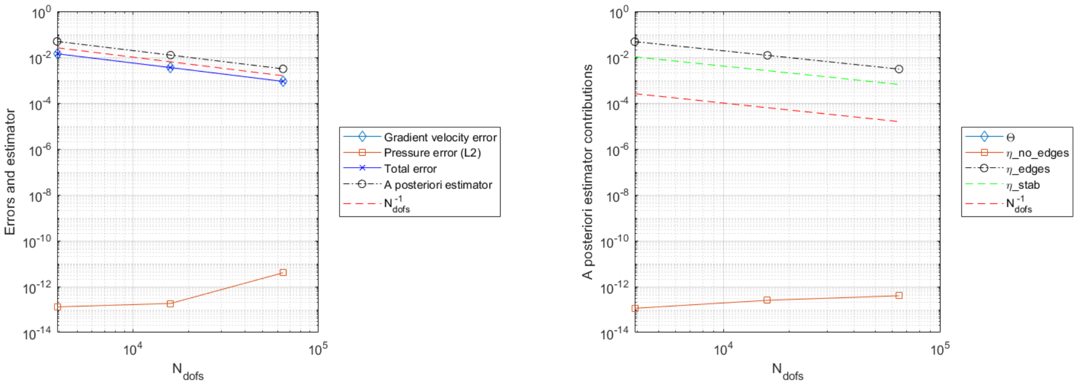
In this case problem is formulated such that solution is:

$$\mathbf{u} = 10 \begin{bmatrix} e^{\left(-\frac{(x+y)^2}{10}\right)} \\ -e^{\left(-\frac{(x+y)^2}{10}\right)} \end{bmatrix} \qquad p = 0 \tag{3.139}$$

On every border non-homogeneous Dirichlet boundary conditions are imposed and $\nu = 1$. The starting mesh is shown in Figure 3.3. Figure 3.10 shows plots of components of velocity solution of VEM problem obtained with the most refined mesh. Table 3.6 and Figure 3.11a report the trend of errors and a posteriori estimators while various contribution in a posteriori estimator can be seen in 3.11b. Errors and a posteriori estimators follow usual trend and the dominant part in a posteriori estimator is given by Θ.

Case 7

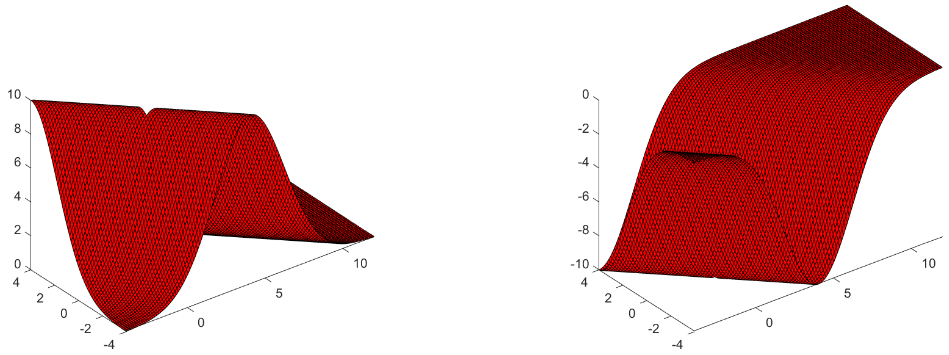
This case is the same as *Case 6*. The only difference is pressure solution which is $p = y^3$ instead of $p = 0$. Therefore term $\mathbf{f}$ is coherently modified. Plots of VEM solution for



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

Figure 3.9: *Case 5* errors and estimators (Stokes problem)



(a) u_x

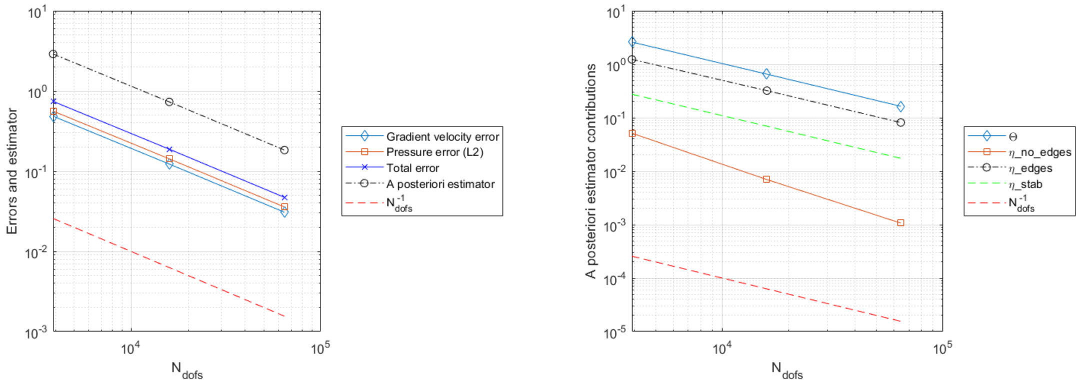
(b) u_y

Figure 3.10: Plots of VEM discrete velocity solution components for case *6* (Stokes problem)

h	gradient error velocity	L^2 error pressure	total error	a posteriori estimator
1/2	0.4848	0.5614	0.7417	2.890
1/4	0.1227	0.1425	0.1880	0.7311
1/8	0.03076	0.03577	0.04717	0.1834

Table 3.6: Gradient errors and a posteriori estimators for uniform refinement sequence: *case 6* (Stokes problem)

velocity are not reported because they are like *case 6*. Figure 3.12a and Table 3.7 report errors and a posteriori estimators. A posteriori estimator contributions are shown in Figure 3.12b. Errors and a posteriori estimators follow a usual trend and the main



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

Figure 3.11: *Case 6* errors and estimators (Stokes problem)

h	gradient error velocity	L^2 error pressure	total error	a posteriori estimator
1/2	0.4848	4.122	4.150	16.17
1/4	0.1227	1.0324	1.040	4.046
1/8	0.03076	0.2582	0.2600	1.012

Table 3.7: Gradient errors and a posteriori estimators for uniform refinement sequence: *case 7* (Stokes problem)

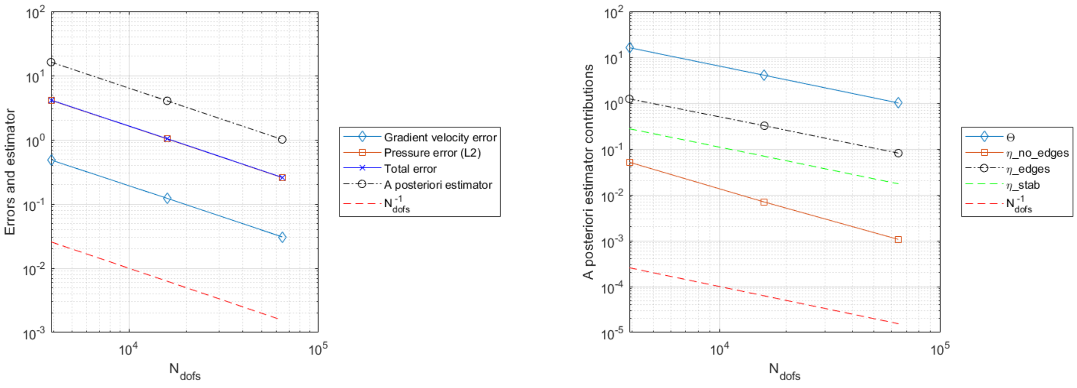
contribution of a posteriori estimator comes from Θ. It is interesting to notice that the dominant error is pressure error while in *Case 6* it was not so dominant. This comes from the use of a pressure solution function which is a polynomial of degree 3 and not zero in this case.

Case 8

In this case data of the problem are chosen such that the solution is:

$$\mathbf{u} = \begin{bmatrix} 4\sin(\pi x)\cos(\pi y) \\ -4\cos(\pi x)\sin(\pi y) \end{bmatrix} \qquad p = 0 \tag{3.140}$$

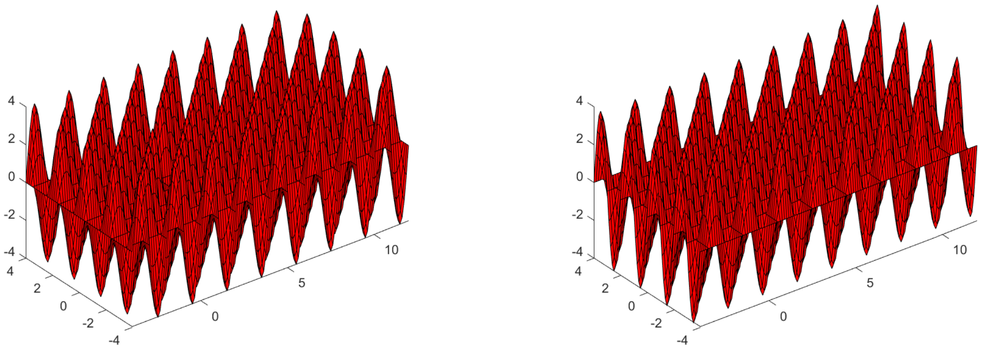
Starting mesh is again shown in Figure 3.3 and $\nu = 1$. Non-homogeneous Dirichlet boundary conditions are imposed on every boundary. It is possible to notice that velocity solution field fulfills incompressibility condition. Figure 3.13 shows plots of components of velocity solution of VEM problem obtained with the most refined mesh. Figure 3.14a and Table 3.8 report errors and a posteriori estimators. A posteriori estimator contributions are shown in Figure 3.14b. Errors here follows a classical trend. Error on velocity is not divided by 4 with the first refinement but this is possible at the beginning of the refinement, in fact the second iteration gives an error which is about 1/4 of previous one as expected. In a posteriori estimator the dominant part is again Θ.



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

Figure 3.12: *Case 7* errors and estimators (Stokes problem)



(a) u_x

(b) u_y

Figure 3.13: Plots of VEM discrete velocity solution components for case *8* (Stokes problem)

Fluid dynamics test case

In fluid dynamics situation described in introduction the explicit form of solution is not known. On walls and on the obstacle no slip conditions are imposed therefore here:

$$\mathbf{u}_{no\_slip} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.141}$$

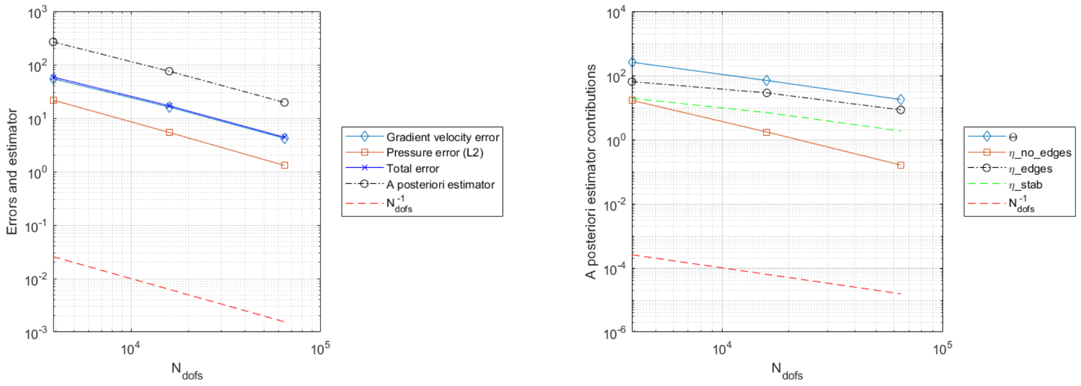
On the outflow boundary there is a Neumann homogeneous condition while on inflow:

$$\mathbf{u}_{inflow} = \begin{bmatrix} -\frac{1}{16}(y-4)(y+4) \\ 0 \end{bmatrix} \tag{3.142}$$

It is clear that the previous condition becomes zero on the walls. The constant factor comes from how the problem is adimensionalized as explained in the next chapter. Gravity force

h	gradient error velocity	L^2 error pressure	total error	a posteriori estimator
1/2	54.7	21.8	58.9	267
1/4	16.2	5.41	17.1	75.5
1/8	4.21	1.31	4.41	19.7

Table 3.8: Gradient errors and a posteriori estimators for uniform refinement sequence: *case 8* (Stokes problem)

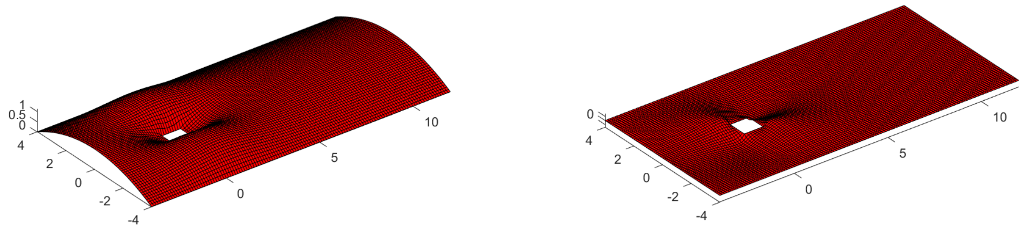


(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

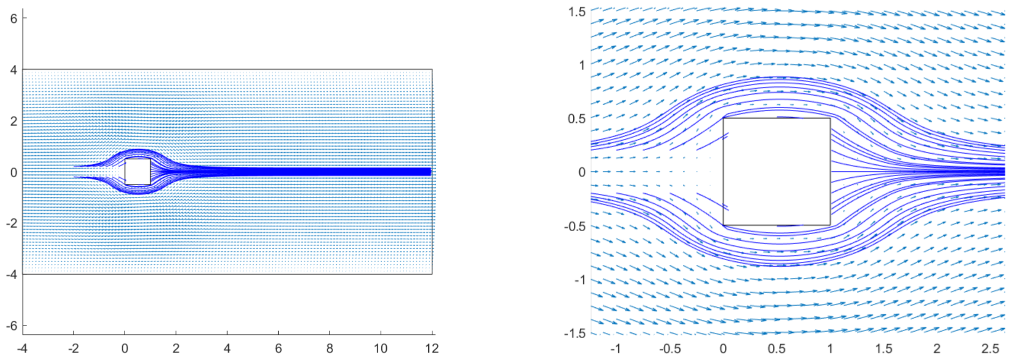
Figure 3.14: *Case 8* errors and estimators (Stokes problem)

is considered irrelevant so $\mathbf{f} = 0$. Viscosity ν is such that $\nu = 1$. Moreover it is possible to notice that incompressibility condition implies $\int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n}\, ds = 0$. The starting mesh is shown in Figure 3.3. Figure 3.15 shows plots of VEM solution components of velocity. Figure 3.16 shows the vector-valued velocity field in the case of the most refined mesh with some streamlines. In Figure 3.17 there are plots of a posteriori estimator and of its various contributions. It is possible to see that a posteriori estimator slope is not optimal. This is probably due to an error which is concentrated around the cylinder so uniform refinement is not the best choice. In fact in most of the domain, function solution is very regular while around cylinder has a more complex form (see Figure 3.15). As shown in Figure 3.16 the fluid flow surrounds the cylinder. The flow right before cylinder is symmetric with respect to the flow right after cylinder. In fact there is no convection and flow is very regular around cylinder thanks to diffusion. Streamlines shows that flow is very similar to Navier-Stokes solution for $Re = 1$ shown in a Figure of Breuer et al. [2000] in fact for $Re \leq 1$, as explained in Breuer et al. [2000], there is no separation of laminar boundary layer which can occur due to convection.



(a) u_x (b) u_y

Figure 3.15: Plots of VEM discrete velocity solution components for fluid dynamics case (Stokes problem)

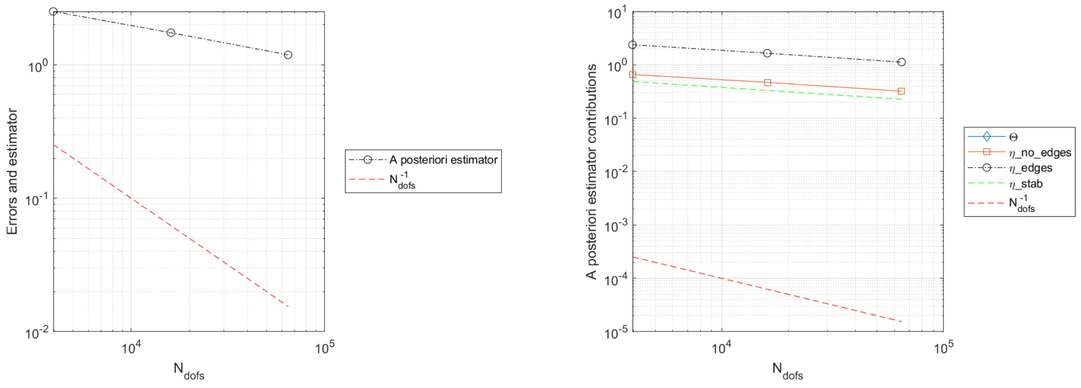


(a) Vector field on the whole domain (b) Vector field around cylinder

Figure 3.16: Velocity field of fluid dynamics case (Stokes uniform refinement). Blue lines are streamlines

3.5.2 Adaptive mesh refinements

With *Adaptive mesh refinement* meshes are refined through **Solve-mark-estimate-refine** procedure. For every case number of refinements and parameter θ are specified. Numeration is the same as uniform mesh refinements. Furthermore some meshes coming from adaptive refinement process are shown. For every case the starting mesh is the one of corresponding case in uniform mesh refinement.



(a) Plot of estimator

(b) Plot of parts of a posteriori estimator

Figure 3.17: *Fluid dynamics case* estimators (Stokes problem)

Case 1

Here 9 refinements are realized with $\theta = 0.5$. Figure 3.18 shows 4 meshes of refinement procedure. Figure 3.19a shows errors and a posteriori estimators while a posteriori estimator contributions can be seen in Figure 3.19b. It is possible to see that refinement is prevalent in places where pressure is higher, moreover refinement follows a symmetric pattern accordingly with the pressure solution which is antisymmetric with respect to the line $y = 0$. Errors and a posteriori estimator follow a regular trend. It is possible to notice that velocity error increases a lot after adding the first hanging nodes. Anyway the total error and a posteriori estimator follow a very similar slope.

Case 2

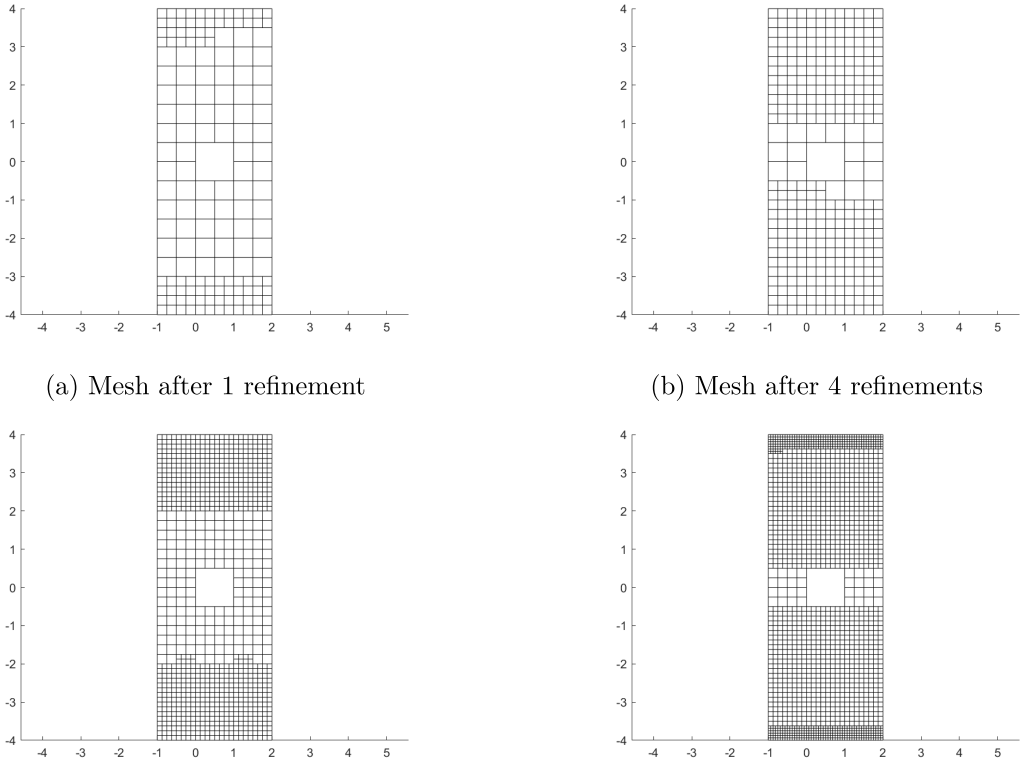
In this case 12 refinement are done with $\theta = 0.4$. Figure 3.20 shows 4 meshes of refinement procedure. Figure 3.21a shows errors and a posteriori estimator while a posteriori estimator contributions can be seen in Figure 3.21b. It is possible to see that all errors and a posteriori estimator shows a usual trend following the slope related to $O(h^2)$ convergence. A posteriori estimator main contribution is given by Θ.

Case 3

Here 12 refinements are realized with $\theta = 0.4$. In this case the sequence of meshes is not shown because it is basically the same as *Case 2*. Figure 3.22a shows errors and a posteriori estimator while a posteriori error contributions can be seen in Figure 3.22b. It is possible to see that the trend is very similar to *Case 2*.

Case 4

Again here 12 refinements are realized with $\theta = 0.4$. In this case the sequence of meshes is not shown because it is basically the same as *Case 2*. Figure 3.23a shows errors and



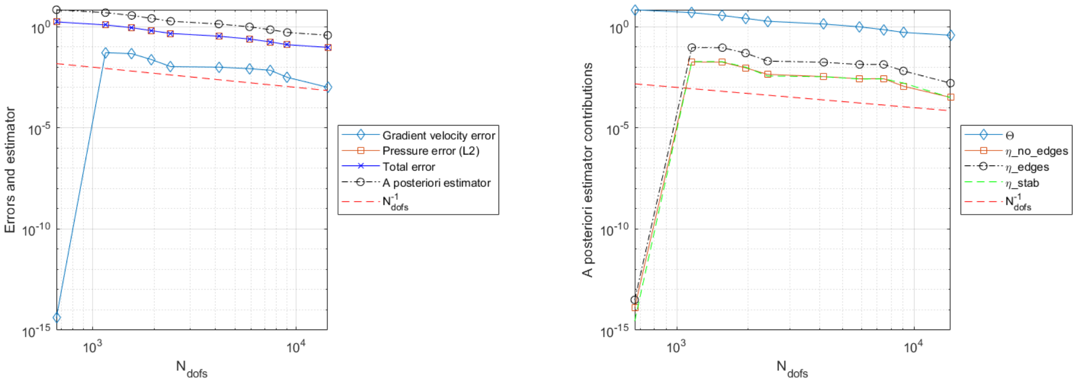
(a) Mesh after 1 refinement

(b) Mesh after 4 refinements

(c) Mesh after 7 refinements

(d) Mesh after 9 refinements

Figure 3.18: 4 meshes generated in adaptive mesh refinement of *case 1* (Stokes problem)

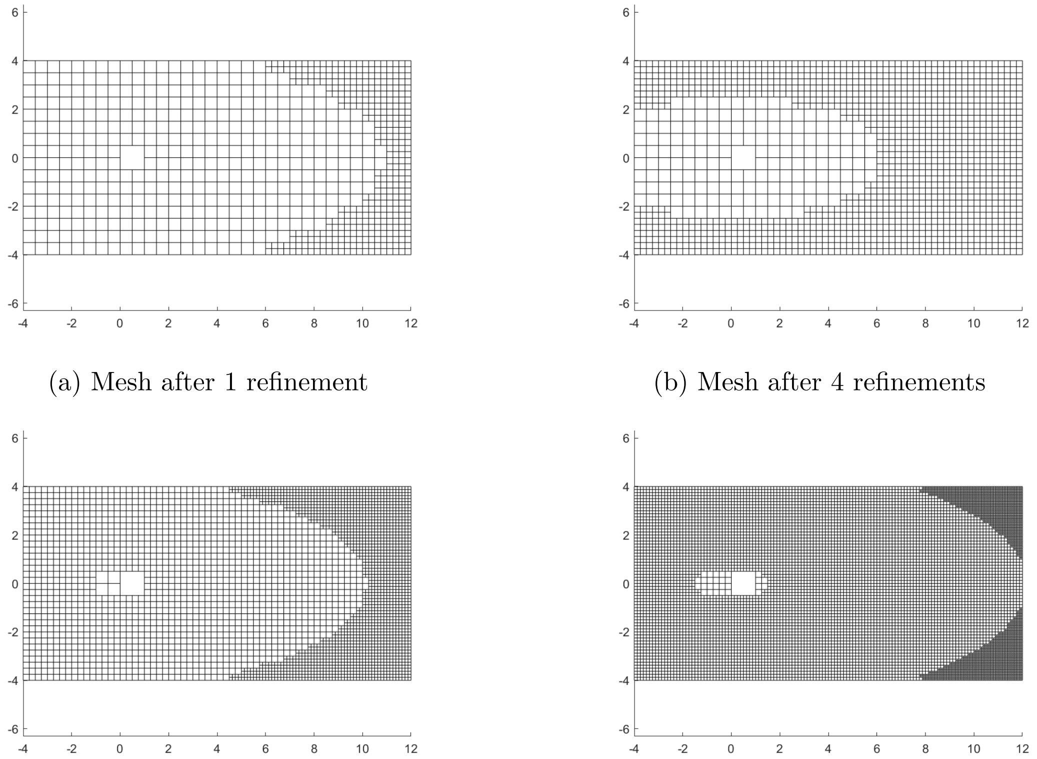


(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

Figure 3.19: *Case 1* errors and estimators (adaptive mesh refinement) (Stokes problem)

a posteriori estimators while a posteriori estimator contributions can be seen in Figure



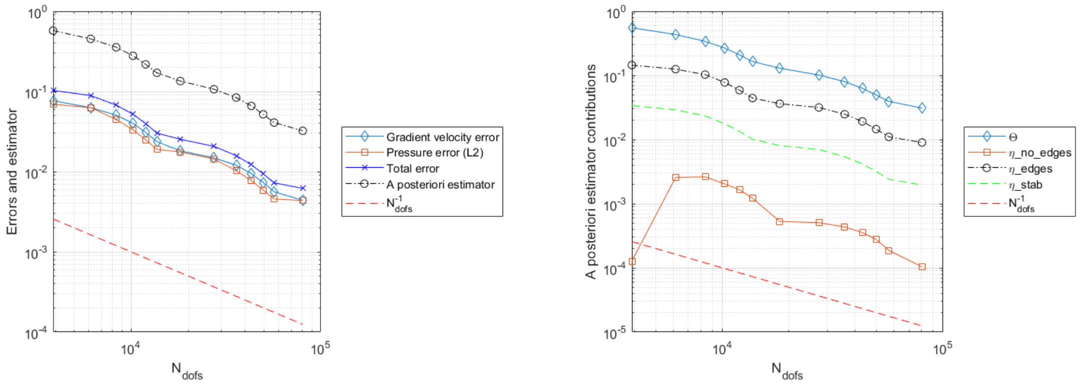
(a) Mesh after 1 refinement

(b) Mesh after 4 refinements

(c) Mesh after 7 refinements

(d) Mesh after 12 refinements

Figure 3.20: 4 meshes generated in adaptive mesh refinement of *case 2* (Stokes problem)

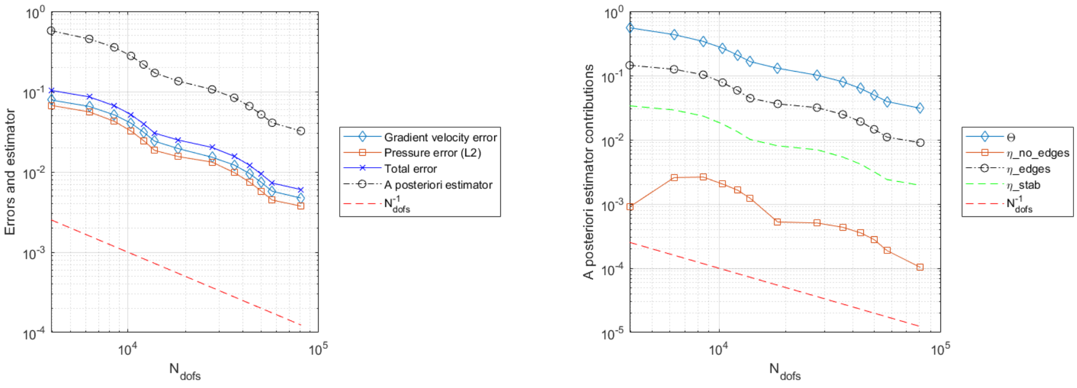


(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

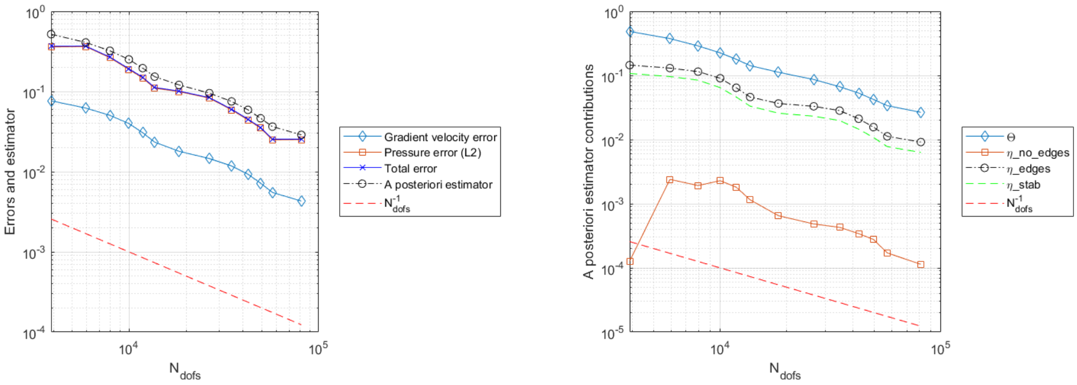
Figure 3.21: *Case 2* errors and estimators (adaptive mesh refinement) (Stokes problem)

3.23b. It is possible to see that the trend is very similar to *Case 2*. Here pressure error



(a) Plot of errors and estimator (b) Plot of parts of a posteriori estimator

Figure 3.22: *Case 3* errors and estimators (adaptive mesh refinement) (Stokes problem)



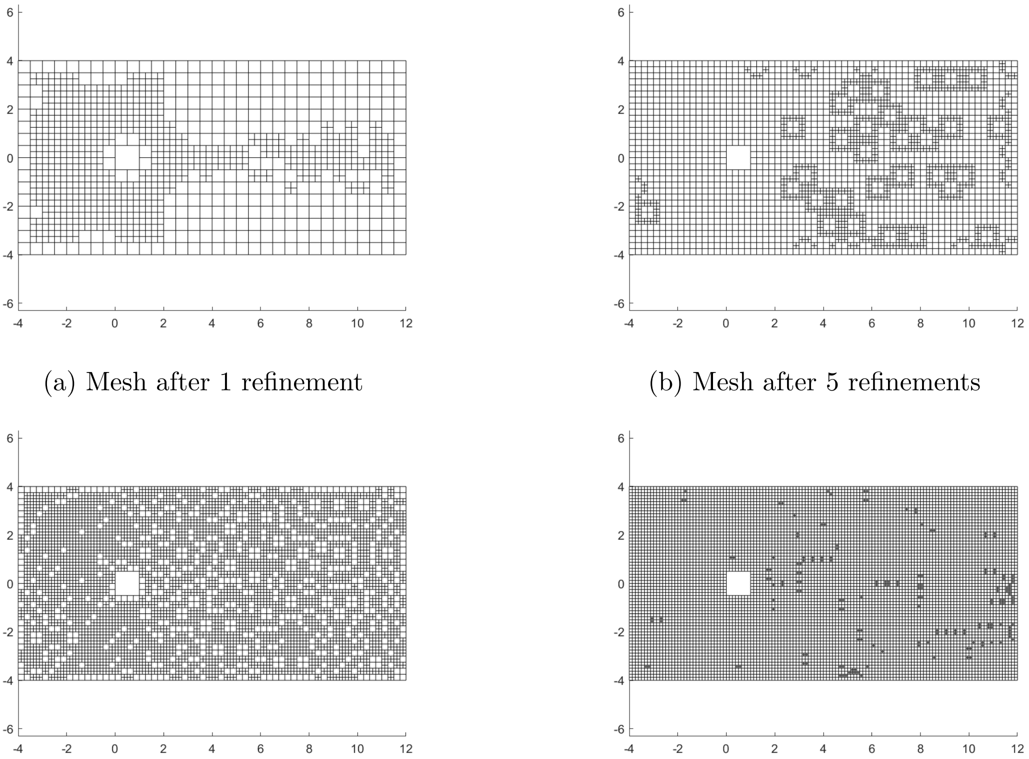
(a) Plot of errors and estimator (b) Plot of parts of a posteriori estimator

Figure 3.23: *Case 4* errors and estimators (adaptive mesh refinement) (Stokes problem)

becomes the dominant error respect to cases *3* and *2*. The trend of a posteriori estimator is very similar to cases *3* and *2*.

Case 5

Here 9 refinements are realized with $\theta = 0.4$. Figure 3.24 shows 4 meshes of refinement procedure. Figure 3.25a shows errors and a posteriori estimators while a posteriori eestimator contributions can be seen in Figure 3.25b. It is possible to see that the dominant error is velocity gradient error in fact its value decreases with refinement while pressure error has not a monotone trend as it does not influence much the total error. Moreover it is possible to see that *η_edges* is the main contribution in a posteriori estimator.



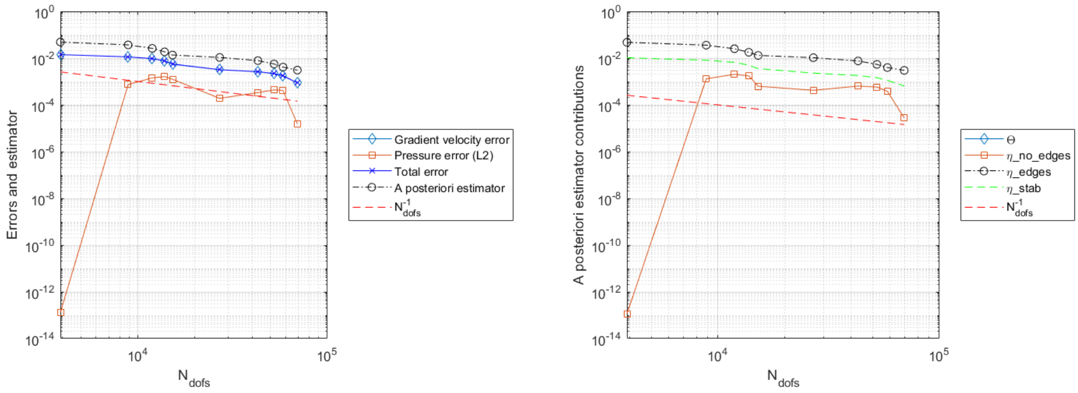
(a) Mesh after 1 refinement

(b) Mesh after 5 refinements

(c) Mesh after 7 refinements

(d) Mesh after 9 refinements

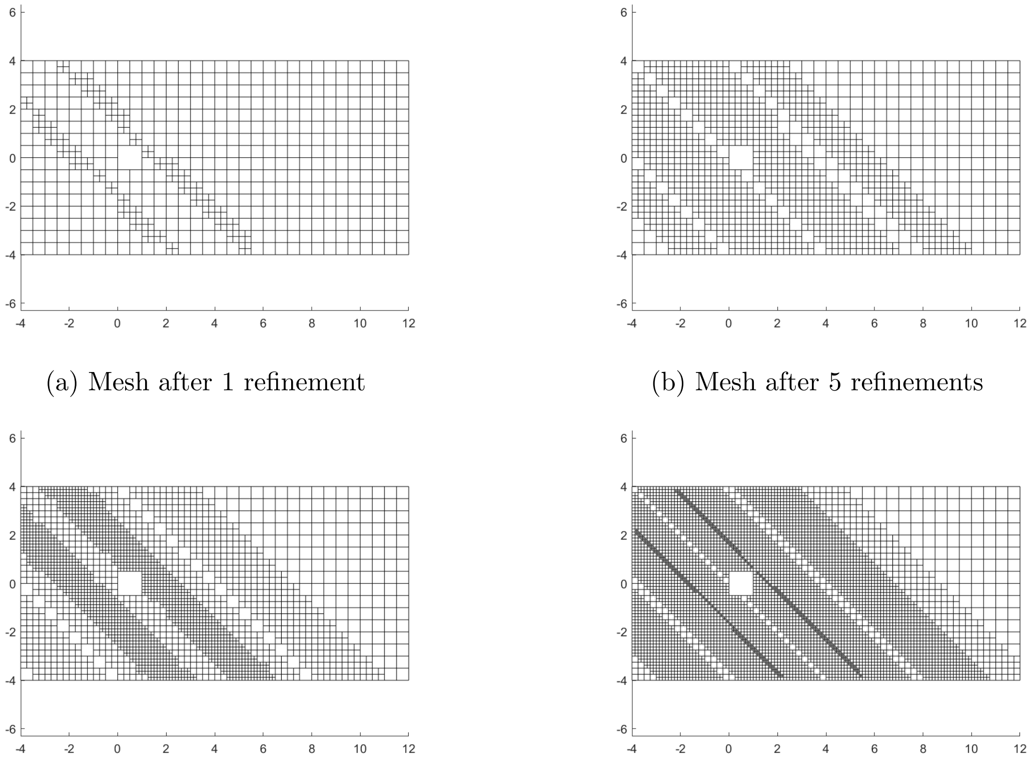
Figure 3.24: 4 meshes generated in adaptive mesh refinement of *case 5* (Stokes problem)



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori error

Figure 3.25: *Case 5* errors and estimators (adaptive mesh refinement) (Stokes problem)



(a) Mesh after 1 refinement

(b) Mesh after 5 refinements

(c) Mesh after 8 refinements

(d) Mesh after 11 refinements

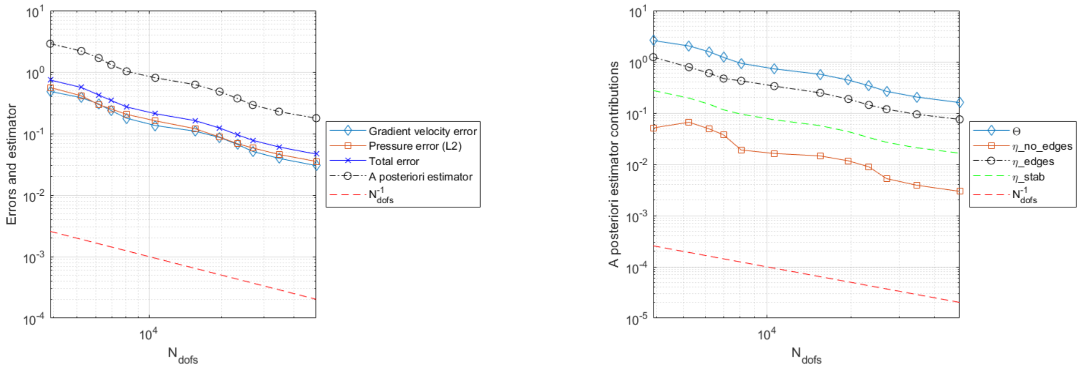
Figure 3.26: 4 meshes generated in adaptive mesh refinement of *case 6* (Stokes problem)

Case 6

In this case 11 refinements are realized with $\theta = 0.4$. Figure 3.26 shows 4 meshes of refinement procedure. Figure 3.27a shows errors and a posteriori estimator while a posteriori estimator contributions can be seen in Figure 3.27b. It is possible to see that refinement follows the shape of the Gaussian-like velocity components functions. The general trend of errors is very regular.

Case 7

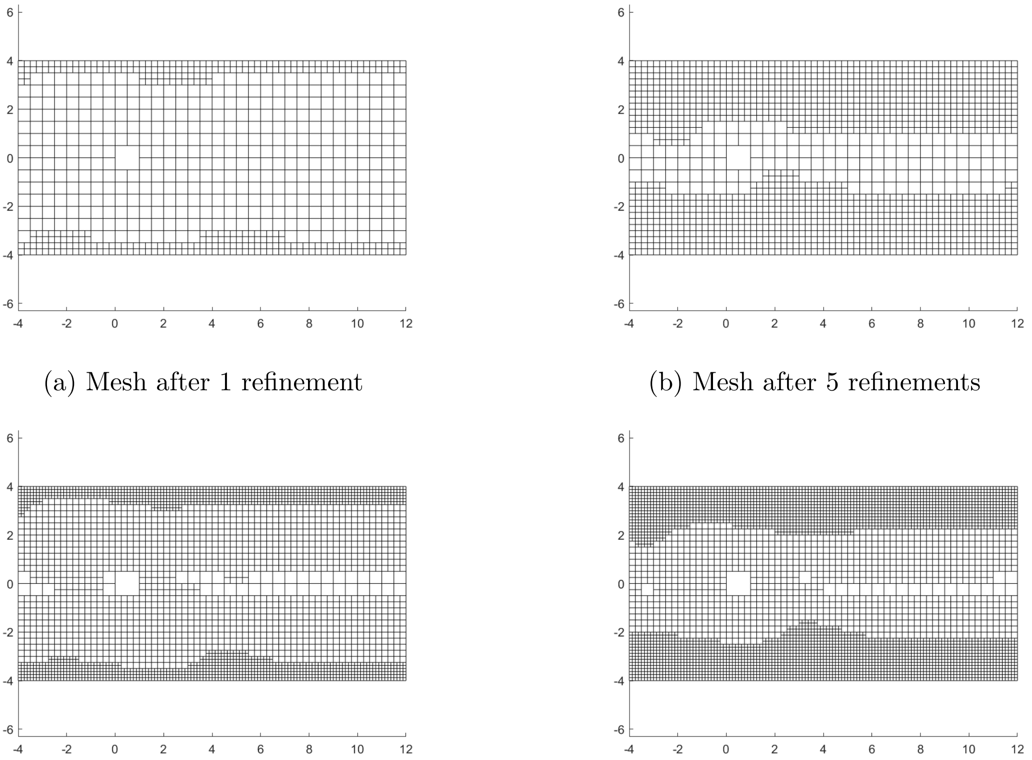
In this case 9 refinements are realized with $\theta = 0.4$. Figure 3.28 shows 4 meshes of refinement procedure. Figure 3.29a shows errors and a posteriori estimators while a posteriori estimator contributions can be seen in Figure 3.29b. It is possible to see that even if velocity solution vector field is the same as *Case 6* the refinement process is completely different. It seems that estimator follows pressure to refine, considering places where pressure has higher values. This makes sense because pressure error, here, becomes much greater than gradient velocity error. It can be seen that (see Figure 3.29a) the total error coincides basically with pressure error. A posteriori estimator follows a regular trend.



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori error

Figure 3.27: *Case 6* errors and estimators (adaptive mesh refinement) (Stokes problem)



(a) Mesh after 1 refinement

(b) Mesh after 5 refinements

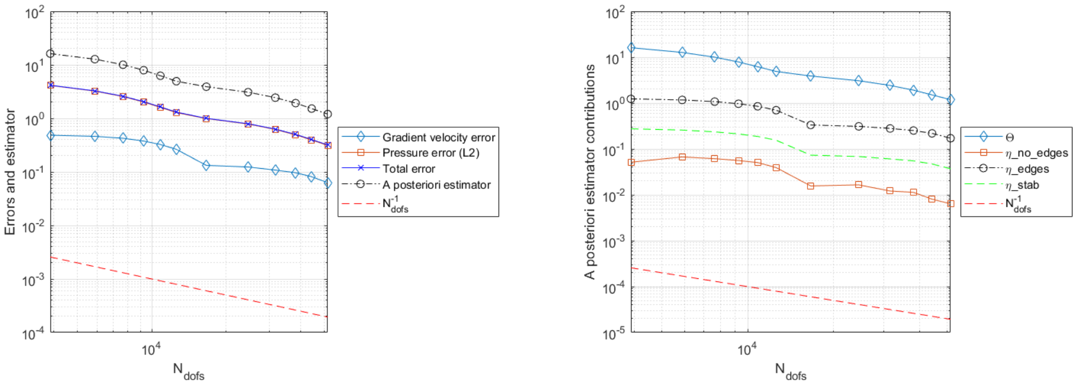
(c) Mesh after 7 refinements

(d) Mesh after 9 refinements

Figure 3.28: 4 meshes generated in adaptive mesh refinement of *case 7* (Stokes problem)

Case 8

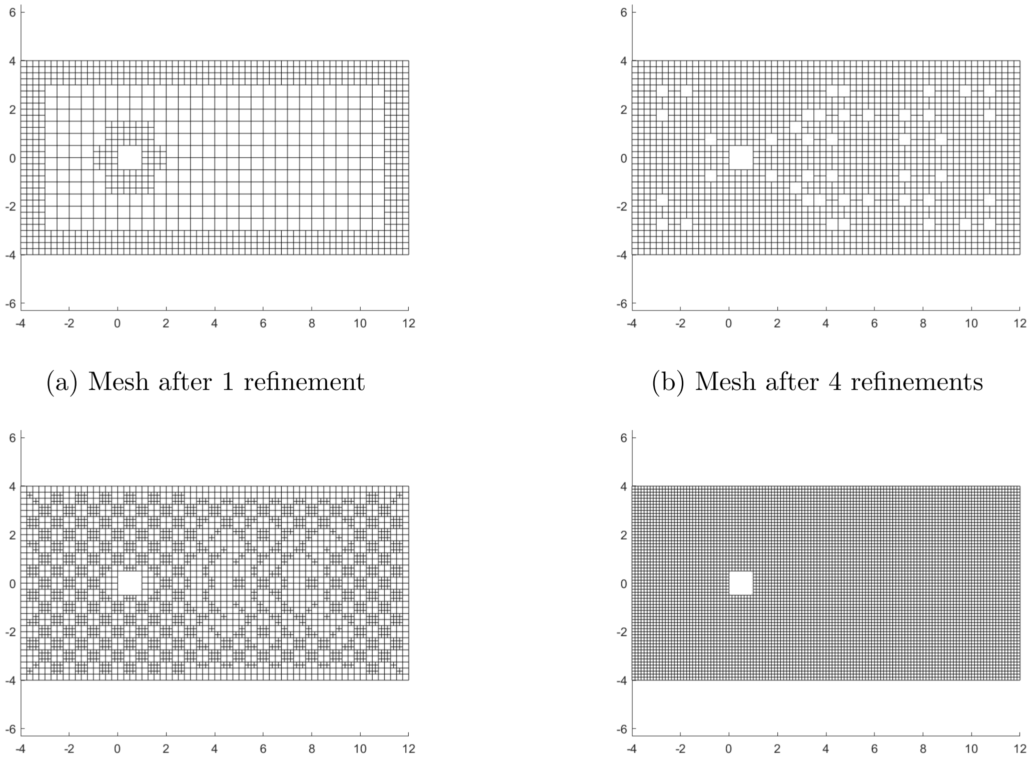
In this case 11 refinements are realized with $\theta = 0.4$. Figure 3.30 shows 4 meshes of refinement procedure. Figure 3.31a shows errors and a posteriori estimators while a posteriori



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

Figure 3.29: *Case 7* errors and estimators (adaptive mesh refinement) (Stokes problem)



(a) Mesh after 1 refinement

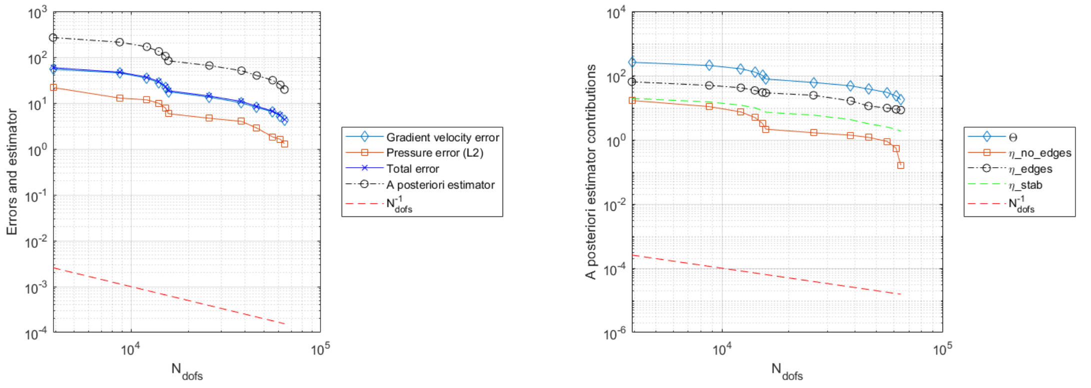
(b) Mesh after 4 refinements

(c) Mesh after 7 refinements

(d) Mesh after 11 refinements

Figure 3.30: 4 meshes generated in adaptive mesh refinement of *case 8* (Stokes problem)

estimator contributions can be seen in Figure 3.31b. It is possible to see also in this case



(a) Plot of errors and estimator

(b) Plot of parts of a posteriori estimator

Figure 3.31: *Case 8* errors and estimators (adaptive mesh refinement) (Stokes problem)

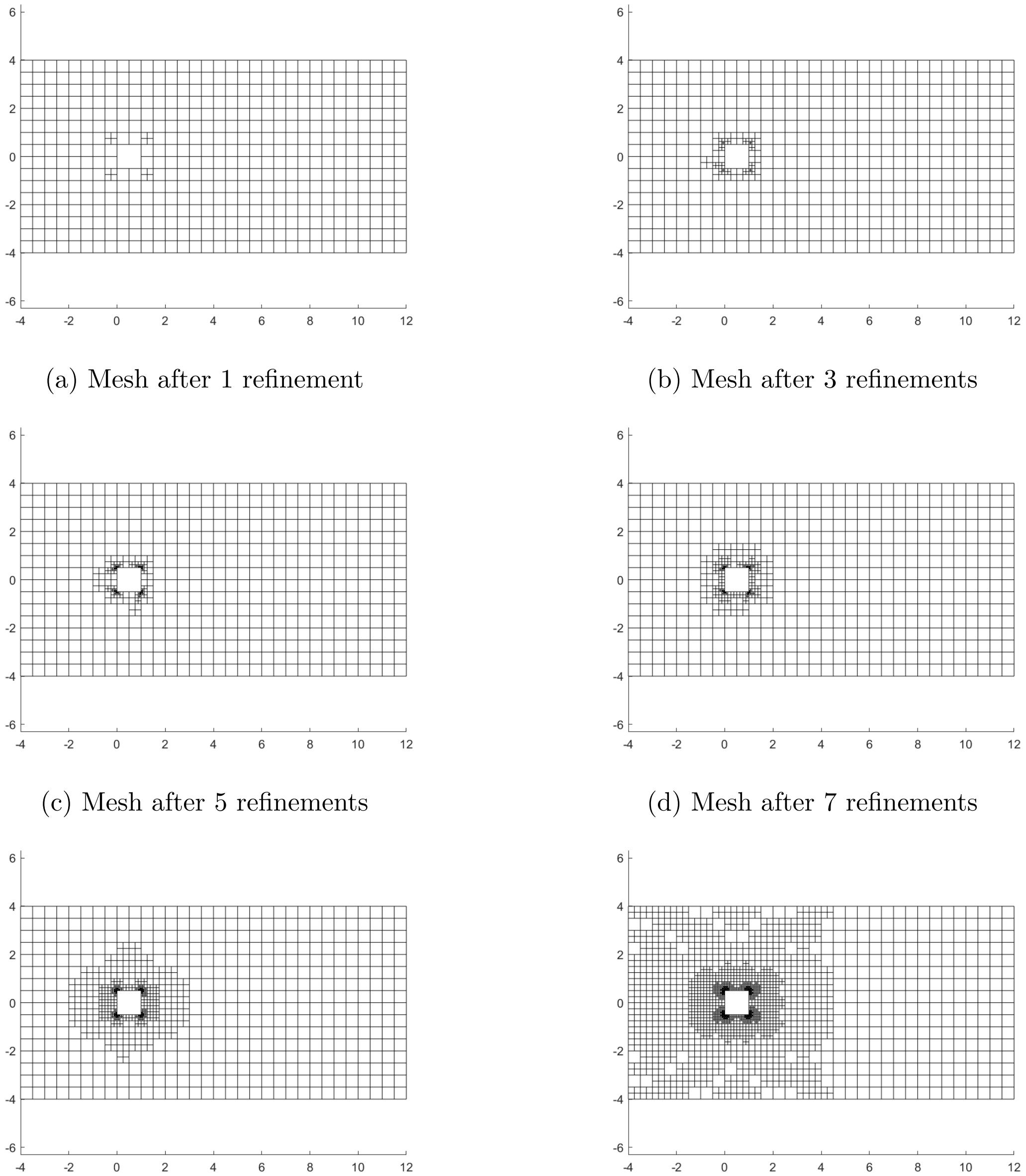
that error and a posteriori estimator follow a classical trend.

An important observation can be done.

Remark 10 *In all the cases a posterior estimator was computed including the term coming from stabilization form. However from the plots shown it is possible to see that stabilization part contribution is not the main relevant part in all the cases taken into account. This fact suggests that a result, similar to the one found in Beirão da Veiga et al. [2021b] related to stabilization part, may be valid for Stokes case. Anyway this is only a conjecture.*

Fluid dynamics test case

Fluid dynamics case refinement is realized with $\theta = 0.4$ with 12 refinement. More refinement steps are not considered because matrices in VEM procedure would become nearly singular due to the presence of very small elements. Figure 3.32 shows some meshes of adaptive refinement. Errors, a posteriori estimator and its contributions are shown in 3.33. It is possible to see that the a posteriori estimator, differently from the corresponding uniform refinement case, decreases a lot at beginning and it tends to become parallel to the slope related to $O(h^2)$ convergence. The same observation can be done for the various contributions of a posteriori estimator. The main contribution comes from η_edges. Especially at the beginning, in fact, the refinement is extremely located around square cylinder particularly at vertices. This makes sense because flow changes a lot its direction with a diffusion (the only effect present in Stokes case) around the cylinder. Moreover refinement looks symmetric both in x and y directions because there are no convective effects. Figure 3.34 shows the vector velocity field for the most refined mesh.



(a) Mesh after 1 refinement

(b) Mesh after 3 refinements

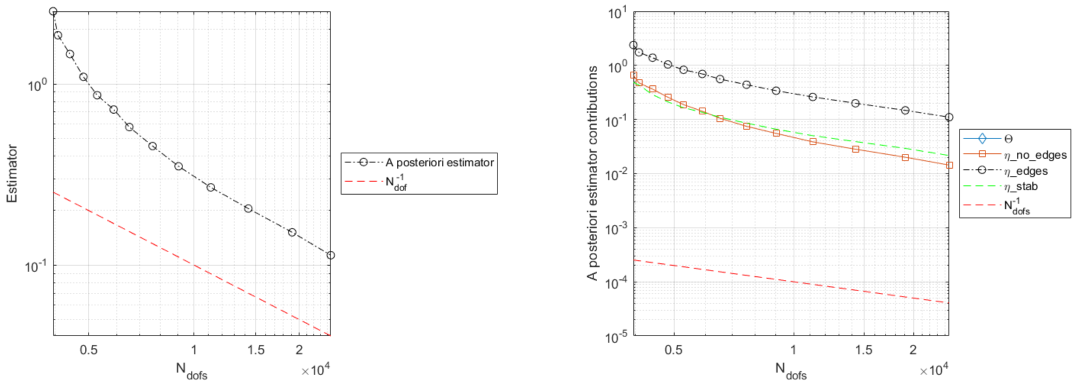
(c) Mesh after 5 refinements

(d) Mesh after 7 refinements

(e) Mesh after 9 refinements

(f) Mesh after 12 refinements

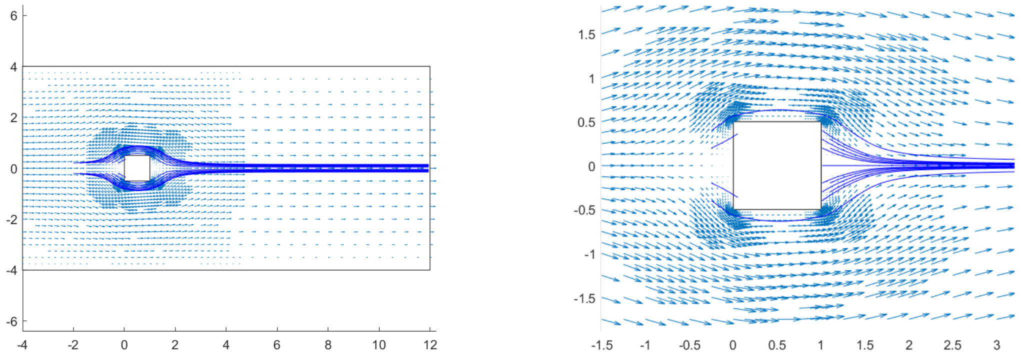
Figure 3.32: 6 meshes generated in adaptive mesh refinement of *Fluid dynamics test case* (Stokes problem)



(a) Plot of estimator

(b) Plot of parts of a posteriori estimator

Figure 3.33: (Stokes) *Fluid dynamics test case* estimators (adaptive mesh refinement)



(a) Plot of the whole mesh

(b) Plot near square cylinder

Figure 3.34: (Stokes) *Fluid dynamics test case* solution velocity field (adaptive mesh refinement). Blue lines are streamlines.

Chapter 4

Solution of the Navier-Stokes problem using the VEM

The Navier-Stokes equation system describes the motion of an incompressible fluid considering also the convection term. The assumptions about the fluid-dynamics made in the Introduction, in particular the limitation given by the use of a low Reynolds number, leads to consider steady Navier Stokes equations. Moreover no stabilization terms are adopted. The whole formulation of the VEM discrete Navier-Stokes problem and most of the structure of next section is taken from the work in Beirão da Veiga et al. [2018]. All numerical tecniques presented have been implemented in *Matlab*.

4.1 The continuous problem

The steady Navier-Stokes problem considered is defined on a polygonal domain $\Omega \subset \mathbb{R}^2$ as follows (see Beirão da Veiga et al. [2018])

$$\text{find} \quad (\mathbf{u}, p) \quad \text{such that} \quad \begin{cases} -\nu\boldsymbol{\Delta}\mathbf{u} + (\nabla\mathbf{u})\mathbf{u} - \nabla p = \mathbf{f} \quad \text{in } \Omega \\ \text{div}\mathbf{u} = 0 \quad \text{in} \quad \Omega \\ \mathbf{u} = \mathbf{0} \quad \text{on} \quad \partial\Omega_D \\ (\nu\nabla\mathbf{u} + p\mathbf{I})\mathbf{n} = \mathbf{0} \quad \text{on} \quad \partial\Omega_N \end{cases} \tag{4.1}$$

where $\mathbf{u}$ and p are the velocity and pressure solutions. $\mathbf{f}$ is the external force. $\text{div}\mathbf{u} = 0$ gives the incompressibility condition like in Stokes problem. $\nu > 0$ is the diffusion coefficient (here chosen constant) and the symbols used have the same meaning as in the case of Stokes problem. The sign of pressure inside equations is chosen as in Beirão da Veiga et al. [2018]. $(\nabla\mathbf{u})\mathbf{u}$ is the convective term: $\nabla\mathbf{u}$ results in a tensor (the Jacobian matrix) and the second $\mathbf{u}$ specifies a "direction" for the gradient, indeed $(\nabla\mathbf{u})\mathbf{u} = (\mathbf{u}\cdot\nabla)\mathbf{u}$.

Like Stokes case the theoretical analysis is done using homogeneous Dirichlet boundary condition but VEM can be applied using Neumann boundary condition (here limited to homogeneous) and Dirichlet non-homogeneous Dirichlet boundary. More details will be written in the implementation part, however general considerations for boundary conditions done for Stokes case still hold.

The problem can be written in variational form in a similar way as the Stokes problem since the only new term which needs particular attention is the convective one. Let now $\mathbf{V}$, Q and Q^N be the spaces defined, respectively, in (3.3), (3.4) and (3.5) with the usual norm (see (3.6)).

Let now be assumed $\mathbf{f} \in [L^2(\Omega)]^2$. Bilinear forms $a(\cdot,\cdot) : \mathbf{V} \times \mathbf{V}$ and $b(\cdot,\cdot) : \mathbf{V} \times Q$ are defined as in Stokes case (see 3.7 and 3.8). A new trilinear form $c(\cdot;\cdot,\cdot) : \mathbf{V} \times \mathbf{V} \times \mathbf{V} \to \mathbb{R}$ is introduced in order to write in variational form the convective term. It is defined as:

$$c(\mathbf{w};\mathbf{u},\mathbf{v}) := \int_\Omega (\nabla\mathbf{u})\mathbf{w} \cdot \mathbf{v}\, d\Omega \quad \forall\, \mathbf{w},\mathbf{u},\mathbf{v} \in \mathbf{V} \tag{4.2}$$

The variational continuous Navier-Stokes problem can be defined as (see Beirão da Veiga et al. [2018]):

$$\text{find } (\mathbf{u},p) \in \mathbf{V} \times Q \text{ such that} \quad \begin{cases} a(\mathbf{u},\mathbf{v}) + c(\mathbf{u};\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) = (\mathbf{f},\mathbf{v}) & \forall\, \mathbf{v} \in \mathbf{V} \\ b(\mathbf{u},q) = 0 \quad \forall\, q \in Q \end{cases} \tag{4.3}$$

with:

$$(\mathbf{f},\mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v}\, d\Omega \tag{4.4}$$

Now the properties of linear forms are listed:

- $a(\cdot,\cdot)$ is a *continuous* and *coercive* bilinear form as defined in 3.10 and 3.12 for Stokes problem.
- $b(\cdot,\cdot)$ is a *continuous* bilinear form (see 3.11)
- $b(\cdot,\cdot)$ satisfies *inf-sup condition* on spaces $\mathbf{V}$ and Q (see 3.13 for definition).
- $c(\cdot;\cdot,\cdot,)$ is continuous. This means:

$$|c(\mathbf{w};\mathbf{u},\mathbf{v})| \le \tilde{C}||\mathbf{w}||_{\mathbf{V}}||\mathbf{u}||_{\mathbf{V}}||\mathbf{v}||_{\mathbf{V}} \quad \forall\, \mathbf{w},\mathbf{u},\mathbf{v} \in \mathbf{V} \tag{4.5}$$

Reporting what is written in Beirão da Veiga et al. [2018], if

$$\gamma := \frac{\tilde{C}||\mathbf{f}||_{-1}}{\nu^2} < 1 \tag{4.6}$$

it is true that variational problem (4.3) has a unique solution $(\mathbf{u},p) \in \mathbf{V}\times$ Q such that

$$||\mathbf{u}||_{\mathbf{V}} \le \frac{||\mathbf{f}||_{H^{-1}}}{\nu} \tag{4.7}$$

As in Stokes case it is possible to rewrite the problem (4.3) in another way (see Beirão da Veiga et al. [2018]) considering space $\mathbf{Z}$ defined as in (3.63).

$$\text{find} \quad \mathbf{u} \in \mathbf{Z}, \quad \text{s.t.} \quad a(\mathbf{u},\mathbf{v}) + c(\mathbf{u};\mathbf{u},\mathbf{v}) = (\mathbf{f},\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{Z} \tag{4.8}$$

It is possible to observe that given $\mathbf{u} \in \mathbf{Z}$ the bilinear form $c(\mathbf{u};\cdot,\cdot) : \mathbf{V} \times \mathbf{V} \to \mathbb{R}$ is skew-symmetric:

$$c(\mathbf{u};\mathbf{w},\mathbf{v}) = -c(\mathbf{u};\mathbf{v},\mathbf{w}) \quad \forall\, \mathbf{w},\mathbf{v} \in \mathbf{V} \tag{4.9}$$

Now it is appropriate to define the trilinear form $\tilde{c}(\cdot;\cdot,\cdot) : \mathbf{V} \times \mathbf{V} \times \mathbf{V} \to \mathbb{R}$:

$$\tilde{c}(\mathbf{w};\mathbf{u},\mathbf{v}) = \frac{1}{2}(c(\mathbf{w};\mathbf{u},\mathbf{v}) - c(\mathbf{w};\mathbf{v},\mathbf{u})) \quad \forall\, \mathbf{u},\mathbf{w},\mathbf{v} \in \mathbf{V} \tag{4.10}$$

4.1.1 Adimesionalization

It is important to make dimensional considerations about the parameters and variables in the Navier-Stokes equations (4.1) assuming that physical quantities are written using *SI units*. Pressure p represents a pressure divided by density ρ which, here, will be considered equal to 1. Dimension of pressure is therefore $[m]*[s]^{-2}$ (m indicates *meter* and s *second*). $\mathbf{f}$ external force is a force per unit of mass, its dimension is $[m]*[s]^{-2}$. ν is the cinematic viscosity coefficient. Now let U be the characteristic dimension of velocity $\mathbf{u}$ and L the length characteristic dimension. Moreover the characteristic dimension of time T is chosen as $T = L/U$. Considering how other physical quantities dimensions are related to fundamental ones (length, time and so velocity dimensions) it is possible to proceed in writing the equations adimensionally. Let now be defined the adimensional quantities:

$$\mathbf{u}^* = \frac{\mathbf{u}}{U} \quad x^* = \frac{x}{L} \quad y^* = \frac{y}{L} \quad p^* = \frac{p}{U^2} \quad \mathbf{f}^* = \frac{\mathbf{f}}{U^2/L} \tag{4.11}$$

where x and y are the cartesian coordinates.
Replacing the dimensional quantities with the new adimensional ones it is possible to write:

$$\begin{cases} -\frac{1}{Re}\mathbf{\Delta}\mathbf{u}^* + (\nabla\mathbf{u}^*)\mathbf{u}^* - \nabla p^* = \mathbf{f}^* \\ \mathrm{div}\mathbf{u}^* = 0 \end{cases} \tag{4.12}$$

where differential operators $\mathbf{\Delta}$,∇,∇ and div are now defined using adimensional coordinates x^* and y^*. Re is the Reynolds number:

$$Re = \frac{U\,L}{\nu} \tag{4.13}$$

Reynolds number indicates the ratio between convective effects and diffusion ones. Mathematically problem (4.12) is equivalent to problem (4.1). In the following notation used in (4.1) will be maintained but symbols now will indicate the adimensional formulation with $\nu = 1/Re$.

4.2 VEM formulation and discrete problem

4.2.1 Discretization

Let be $\{\mathcal{T}_h\}_h$ a sequence of decompositions of domain Ω in polygonal elements. Geometrical assumptions 3 done for Stokes case are again assumed (Beirão da Veiga et al. [2018]).

As declared in other chapters here elements are rectangles (from a geometrical point of view) with possibility of hanging nodes (limited by Assumption 2). For those elements assumptions are satisfied. For an element K (x_K, y_K) are coordinates of centroid and h_K is the diameter (3.15). For h it holds definition (3.15).

As for Stokes problem here the polynomial accuracy k chosen is $k = 2$ and definitions will be written restricting to this particular case. Given an element K and $k \in \mathbb{N}^+$ the spaces $\mathbb{P}_k(K)$, $\mathbb{B}_k(K)$ and $\mathcal{G}_k(K)$ are defined in the same way as in Stokes problem (see for the last two ones, respectively, (3.16) and (3.17)). The space $\mathcal{G}^\perp(K)$, following the choice in Beirão da Veiga et al. [2018] is replaced by:

$$\mathcal{G}_k^{\oplus}(K) := \mathbf{x}^\perp[\mathbb{P}_{k-1}(K)] \subseteq [\mathbb{P}_k(K)]^2 \quad \text{where} \quad \mathbf{x}^\perp := (y, -x) \tag{4.14}$$

As remarked in Beirão da Veiga et al. [2017] this change of space is perfectly allowed.

For the definition of spaces and discretization of linear forms it is important to define for a generic k the L^2 projector on an element K. For scalar functions $\Pi_k^{0,K} : L^2(\Omega) \to \mathbb{P}_k(K)$ is defined as Beirão da Veiga et al. [2018]:

$$\int_K q_k(v - \Pi_k^{0,K} v)\, dK = 0 \quad \forall\, v \in L^2(K) \quad \text{and} \quad \forall\, q_k \in \mathbb{P}_k(K) \tag{4.15}$$

The definition is extended to vectorial case $\boldsymbol{\Pi}_k^{0,K} : [L^2(\Omega)]^2 \to [\mathbb{P}_k(K)]^2$ and tensorial one $\boldsymbol{\Pi}_k^{0,K} : [L^2(\Omega)]^{2\times 2} \to [\mathbb{P}_k(K)]^{2\times 2}$. This projection was already used in Stokes case with $k = 0$.

Now, following Beirão da Veiga et al. [2018], the discrete VEM space is chosen differently from Stokes case. In fact an **enhanced virtual element discrete space** is adopted in order to compute L^2 projection $\Pi_2^0(K)$ exactly from degrees of freedom on K. This is useful to discretize the trilinear form related to convection.

Now, always considering $k = 2$, the local VEM space used for Stokes is enlarged in the following way:

$$\mathbf{U}_h^K := \Big\{ \mathbf{v} \in [H^1(K)]^2 \quad \text{s.t.} \quad \mathbf{v}_{|\partial K} \in [\mathbb{B}_2(\partial K)]^2, \; \begin{cases} -\boldsymbol{\Delta}\mathbf{v} - \nabla s \in \mathcal{G}_2^{\oplus}(K), \\ \operatorname{div}\mathbf{v} \in \mathbb{P}_1(K), \end{cases} \\ \text{for some} \quad s \in L^2(K) \Big\} \tag{4.16}$$

Then the discrete enhanced VEM $\mathbf{V}_h^K$ is defined restricting $\mathbf{U}_h^K$ as follows:

$$\mathbf{V}_h^K := \Big\{ \mathbf{v} \in \mathbf{U}_h^K \quad \text{s.t.} \quad \left(\mathbf{v} - \boldsymbol{\Pi}_2^{\nabla,K}\mathbf{v}, \mathbf{g}_2^\perp \right)_{[L^2(K)]^2} = 0 \quad \forall\, \mathbf{g}_2^\perp \in \mathcal{G}_2^{\oplus}(K)/\mathcal{G}_0^{\oplus}(K) = \mathcal{G}_2^{\oplus}(K) \} \tag{4.17}$$

where $\boldsymbol{\Pi}_2^{\nabla,K}$ is defined as in Stokes case (see (3.45)), $\mathcal{G}_2^{\oplus}(K)/\mathcal{G}_0^{\oplus}(K)$ represents polynomials that are in $\mathcal{G}_2^{\oplus}(K)$ and L^2 orthogonal to polynomials in $\mathcal{G}_0^{\oplus}(K)$. However the last space has dimension equal to zero, therefore the space is equivalent to $\mathcal{G}_2^{\oplus}(K)$.

As for the case of Stokes the dimension of the space $\mathbf{V}_h^K$ can be computed (Beirão da Veiga et al. [2018]):

$$\begin{aligned} \dim(\mathbf{V}_h^K) = \dim([\mathbb{B}_2(\partial K])^2 + \dim(\mathcal{G}_0^{\oplus}(K)) + (\dim(\mathbb{P}_1(K)) - 1) = \\ 2n_V 2 + 0 + 3 - 1 = 4n_V + 2 \end{aligned} \tag{4.18}$$

where n_V is the number of vertices (and so of edges) of the polygon.

Degrees of freedom Now degrees of freedom can be defined looking at Beirão da Veiga et al. [2018]. They are the same as Stokes case and also in this case are split in three groups. Then the set of $Dofs$ for $\mathbf{V}_h^K$ is:

1. $Dv1$: values of $\mathbf{v}$ in the vertices of polygon K
2. $Dv2$: values of $\mathbf{v}$ in the medium points of edges of polygon K
3. $Dv3$: the moments up to order 1 and greater than zero of the divergence of $\mathbf{v}$ in K:

$$\int_K (\mathrm{div}\mathbf{v})\, q_1\, dK \quad \forall q_1 \in \mathbb{P}_1(K)/\mathbb{R} \tag{4.19}$$

$\mathbb{P}_1(K)/\mathbb{R}$ means all polynomials of degree ≤ 1 without the constants. $Dv1$ and $Dv2$ are related to values on the boundary of element K and so they describe $\mathbb{B}_2(K)$, $Dv3$ is linked to the fact that in the definition of VEM discrete space $\mathrm{div}\,\mathbf{v} \in \mathbb{P}_1(K)$.

Now an important proposition about computation of projections is stated as reported in Beirão da Veiga et al. [2018].

Proposition 8 *The degrees of freedom Dv allow to compute the following operators exactly:*

$$\mathbf{\Pi}_2^{\nabla,K} : \mathbf{V}_h^K \to [\mathbb{P}_2(K)]^2, \quad \mathbf{\Pi}_2^{0,K} : \mathbf{V}_h^K \to [\mathbb{P}_2(K)]^2, \quad \mathbf{\Pi}_1^{0,K} : \nabla(\mathbf{V}_h^K) \to [\mathbb{P}_1(K)]^{2\times 2} \tag{4.20}$$

The proof of this proposition can be found in Beirão da Veiga et al. [2017] for Nabla operator $\mathbf{\Pi}_2^{\nabla,K}$ and in Vacca [2017] for L^2 projections. The proof in Vacca [2017] is particularly important because it explains how to compute $\mathbf{\Pi}_2^{0,K}$.

It is possible to observe that (Beirão da Veiga et al. [2018]):

Remark 11 *Given any polynomial q_k, where $k \in \mathbb{N}$ is polynomial degree, and any $\mathbf{v} \in \mathbf{V}_h^K$ it is possible to compute:*

$$\int_K \nabla q_k \cdot \mathbf{v}\, dK \tag{4.21}$$

In fact:

$$\int_K \nabla q_k \cdot \mathbf{v}\, dK = \int_{\partial K} q_k\, \mathbf{v} \cdot \mathbf{n}\, ds - \int_K q_k\, \mathrm{div}\mathbf{v}\, dK \tag{4.22}$$

The first integral can be computed using $Dv1$ and $Dv2$. The second one can be computed dividing q_k as $q_k = q_1 + q_1^{\perp}$, where $q_1 \in \mathbb{P}_1(K)$ and $q_1^{\perp}$ is an element L^2 orthogonal to all polynomials in $\mathbb{P}_1(K)$. Therefore the integral with $q_1^{\perp}$ will be equal to 0 because $\mathrm{div}\mathbf{v} \in \mathbb{P}_1(K)$ from definition. It remains:

$$\int_K q_1\, \mathrm{div}\mathbf{v}\, dK \tag{4.23}$$

which is computable using $Dv3$.

For what concerns pressures, as in Stokes, the local VEM space is defined as:

$$Q_h^K = \mathbb{P}_1(K) \tag{4.24}$$

So:

$$\mathrm{dim} Q_h^K = \mathrm{dim}\mathbb{P}_1(K) = 3 \tag{4.25}$$

The local degrees of freedom for Q_h^K are the same as Stokes:

- DvQ: given $q \in Q_h^K$ the moments up to order 1:

$$\int_K q\, p_1 \, dK \quad \forall p_1 \in \mathbb{P}_1(K) \tag{4.26}$$

Global spaces are defined in the following way:

$$\mathbf{V}_h := \{\mathbf{v} \in [H_0^1(\Omega)]^2 \quad \text{s.t} \quad \mathbf{v}_{|K} \in \mathbf{V}_h^K \quad \forall\, K \in \mathcal{T}_h\} \tag{4.27}$$

Using only Dirichlet condition the space for pressure is:

$$Q_h := \{q \in L_0^2(\Omega) \quad \text{s.t} \quad q_K \in Q_h^K \quad \forall K \in \mathcal{T}_h\} \tag{4.28}$$

if there are Neumann condition the space will be:

$$Q_h^N := \{q \in L^2(\Omega) \quad \text{s.t} \quad q_K \in Q_h^K \quad \forall K \in \mathcal{T}_h\} \tag{4.29}$$

So:

$$\dim(\mathbf{V}_h) = 2n_P + 2n_V + 2n_E \tag{4.30}$$

where n_V and n_E are the number of points and edges not on Dirichlet boundary; n_P is the number of elements. Then

$$\dim(Q_h) = 3n_P - 1 \tag{4.31}$$

and

$$\dim(Q_h^N) = 3n_P \tag{4.32}$$

Moreover in the same way as Stokes case the following property holds (see Beirão da Veiga et al. [2018]):

$$\mathrm{div}\mathbf{V}_h \subseteq Q_h \tag{4.33}$$

Multilinear forms

For what concern $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ their discrete formulation is the same as for Stokes case and so it will not be discussed.

Trilinear form The new part is related to trilinear form $c_h(\cdot;\cdot,\cdot) : \mathbf{V}_h \times \mathbf{V}_h \times \mathbf{V}_h \to \mathbb{R}$ coming from convective non-linear term. Given an element $K \in \mathcal{T}_h$ and using the definition in Beirão da Veiga et al. [2018] the local discrete trilinear form is:

$$c_h^K(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) := \int_K \left(\left(\mathbf{\Pi}_1^{0,K} \nabla \mathbf{u}_h \right) \left(\mathbf{\Pi}_2^{0,K} \mathbf{w}_h \right) \right) \cdot \mathbf{\Pi}_2^{0,K} \mathbf{v}_h \, dK \quad \forall\, \mathbf{w}_h, \mathbf{u}_h, \mathbf{v}_h \in \mathbf{V}_h \tag{4.34}$$

Therefore the global discrete trilinear form will be:

$$c_h(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) := \sum_{K \in \mathcal{T}_h} c_h^K(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) \quad \forall\, \mathbf{w}_h, \mathbf{u}_h, \mathbf{v}_h \in \mathbf{V}_h \tag{4.35}$$

Operators in discrete trilinear form c_h^K are computable as already shown, in particular it is possible to compute $\mathbf{\Pi}_2^{0,K}$ thanks to the choice of an enhanced space. Furthermore, as observed in Beirão da Veiga et al. [2018], $c_h(\cdot;\cdot,\cdot)$ is extendable to $\mathbf{V}$ using in the definition $\mathbf{w}, \mathbf{u}, \mathbf{v} \in \mathbf{V}$. Moreover the following proposition (stated and proved in Beirão da Veiga et al. [2018]) holds.

Proposition 9 *Let:*

$$\hat{C}_h := \sup_{\mathbf{w},\mathbf{u},\mathbf{v} \in \mathbf{V}} \frac{|c_h(\mathbf{w}; \mathbf{u}, \mathbf{v})|}{||\mathbf{w}||_{\mathbf{V}}||\mathbf{u}||_{\mathbf{V}}||\mathbf{v}||_{\mathbf{V}}} \tag{4.36}$$

Then $\hat{C}_h$ is uniformly bounded. In other words the trilinear form $c_h(\cdot;\cdot,\cdot)$ is uniformly continuous with respect of h, namely, it holds that:

$$c_h(\mathbf{w}, \mathbf{u}, \mathbf{v}) \leq \hat{C}_h ||\mathbf{u}||_{\mathbf{V}}||\mathbf{w}||_{\mathbf{V}}||\mathbf{v}||_{\mathbf{V}} \tag{4.37}$$

where constant $\hat{C}_h$ does not depend on h

In Beirão da Veiga et al. [2018], $c_h(\cdot;\cdot,\cdot)$ is replaced by the local skew-symmetric trilinear form. Locally, $\tilde{c}_h^K(\cdot;\cdot,\cdot) : \mathbf{V} \times \mathbf{V} \times \mathbf{V} \to \mathbb{R}$ is defined as:

$$\tilde{c}_h^K(\mathbf{w}; \mathbf{u}, \mathbf{v}) := \frac{1}{2} c_h^K(\mathbf{w}; \mathbf{u}, \mathbf{v}) - \frac{1}{2} c_h^K(\mathbf{w}; \mathbf{v}, \mathbf{u}) \quad \forall \mathbf{w}, \mathbf{u}, \mathbf{v} \in \mathbf{V} \tag{4.38}$$

Globally:

$$\tilde{c}_h(\mathbf{w}; \mathbf{u}, \mathbf{v}) := \sum_{K \in \mathcal{T}_h} \tilde{c}_h^K(\mathbf{w}; \mathbf{u}, \mathbf{v}) \quad \forall \mathbf{w}, \mathbf{u}, \mathbf{v} \in \mathbf{V} \tag{4.39}$$

This trilinear form is used for the theoretical part reporting results in Beirão da Veiga et al. [2018]. However, as remarked below, here in the implementation $c_h^K(\cdot;\cdot,\cdot)$ is used.

External force approximation The right-hand side is computed differently with respect to Stokes case, in fact $\mathbf{f}$ is approximated by (see Beirão da Veiga et al. [2018]):

$$\mathbf{f}_h := \mathbf{\Pi}_2^{0,K} \mathbf{f} \quad \forall K \in \mathcal{T}_h \tag{4.40}$$

Therefore, given $\mathbf{v}_h \in \mathbf{V}_h$:

$$(\mathbf{f}_h, \mathbf{v}_h) = \sum_{K \in \mathcal{T}_h} \int_K \mathbf{f}_h \cdot \mathbf{v}_h \, dK = \sum_{K \in \mathcal{T}_h} \int_K \mathbf{\Pi}_2^{0,K} \mathbf{f} \cdot \mathbf{v}_h \, dK = \sum_{K \in \mathcal{T}_h} \int_K \mathbf{f} \cdot \mathbf{\Pi}_2^{0,K} \mathbf{v}_h \, dK \tag{4.41}$$

$\mathbf{\Pi}_2^{0,K} \mathbf{v}_h$ is computable as already said, so the integral can be computed with a quadrature formula as discussed in the implementation part.

4.2.2 Discrete problem

Now, following Beirão da Veiga et al. [2018], it is possible to define the discrete problem:

$$\text{find} \quad (\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h(Q_h^N), s.t.$$
$$\begin{cases} a_h(\mathbf{u}_h, \mathbf{v}_h) + \tilde{c}_h(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = (\mathbf{f}_h, \mathbf{v}_h) \quad \forall\, \mathbf{v}_h \in \mathbf{V}_h \\ b(\mathbf{u}_h, q_h) = 0 \quad \forall\, q_h \in Q_h \end{cases} \tag{4.42}$$

Remark 12 *Even if the theoretical part is done reporting results in Beirão da Veiga et al. [2018] using trilinear form $\tilde{c}_h(\cdot;\cdot,\cdot)$, here, we use the trilinear form $c_h(\cdot;\cdot,\cdot)$. Therefore the discrete problem is*

$$\mathit{find} \quad (\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h(Q_h^N), s.t.$$
$$\begin{cases} a_h(\mathbf{u}_h, \mathbf{v}_h) + c_h(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = (\mathbf{f}_h, \mathbf{v}_h) \quad \forall\, \mathbf{v}_h \in \mathbf{V}_h \\ b(\mathbf{u}_h, q_h) = 0 \quad \forall\, q_h \in Q_h \end{cases} \tag{4.43}$$

So, in the implementation part, $c_h(\cdot;\cdot,\cdot)$ will be always used.

Symmetry and stability of $a_h(\cdot,\cdot)$ implies that it is continuous and coercive with respect to $\mathbf{V}$-norm (Beirão da Veiga et al. [2018]). Moreover using previous results in Beirão da Veiga et al. [2017] the following proposition written in Beirão da Veiga et al. [2018] holds:

Proposition 10 *Given the discrete space $\mathbf{V}_h$ and Q_h, there exists a constant $\hat{\beta}$ independent of h*

$$\sup_{\mathbf{v}_h \in \mathbf{V}_h\, \mathbf{v}_h \neq \mathbf{0}} \frac{b(\mathbf{v}_h, q_h)}{||\mathbf{v}_h||_{\mathbf{V}}} \geq \hat{\beta}||q_h||_Q \quad \forall q_h \in Q_h \tag{4.44}$$

This implies with property (4.33):

$$div \mathbf{V}_h = Q_h \tag{4.45}$$

The problem (4.42) is well-posed thanks to coercivity property of $a_h(\cdot,\cdot)$, the skew-symmetry of $\tilde{c}_h(\cdot;\cdot,\cdot)$, and inf-sup condition (4.44) (see Beirão da Veiga et al. [2018]).

The following theorem stated in Beirão da Veiga et al. [2018] holds (here it is extended also with the use of space Q_h^N):

Theorem 8 *Assuming that*

$$\gamma_h := \frac{\hat{C}_h||\mathbf{f}_h||_{-1}}{\alpha_*^2\nu^2} \leq r < 1 \tag{4.46}$$

the discrete VEM Navier-Stokes problem (4.42) has a unique solution $(\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h$, such that

$$||\mathbf{u}_h||_{\mathbf{V}} \leq \frac{||\mathbf{f}_h||_{-1}}{\alpha_*\nu} \tag{4.47}$$

Using spaces defined for Stokes $\mathbf{Z}$ (3.63) and $\mathbf{Z_h}$ (3.64) with property:

$$\mathbf{Z}_h \subseteq \mathbf{Z} \tag{4.48}$$

the discrete problem (4.42) can be reformulated:

$$\text{find} \quad \mathbf{u}_h \in \mathbf{Z}_h, s.t. \quad a_h(\mathbf{u}_h, \mathbf{v}_h) + \tilde{c}_h(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h) = (\mathbf{f}_h, \mathbf{v}_h) \quad \forall\, \mathbf{v}_h \in \mathbf{Z}_h \tag{4.49}$$

Interpolation analysis

In Beirão da Veiga et al. [2018] interpolation estimate for enhanced space $\mathbf{V}_h$ was proved. Here the theorem is reported.

Theorem 9 *Let $\mathbf{v} \in H^{s+1} \cap \mathbf{V}$ for $0 < s \leq 2$. Then there exists $\mathbf{v}_I \in \mathbf{V}_h$ such that*

$$||\mathbf{v} - \mathbf{v}_I||_0 + h||\mathbf{v} - \mathbf{v}_I||_{\mathbf{V}} \leq Ch^{s+1}|\mathbf{v}|_{s+1}, \tag{4.50}$$

where C is constant depending only on method order ($k = 2$) and constants in geometrical assumptions (see Assumptions 3).

Convergence analysis

The convergence theorem for VEM Navier-Stokes problem is derived in Beirão da Veiga et al. [2018] using some Lemmas. Here some of those lemmas are reported, for proofs of some lemmas and detailed derivation see the article Beirão da Veiga et al. [2018]. First of all a classical approximation result is reported (see Brenner and Scott [2008]):

Lemma 2 *Let $K \in \mathcal{T}_h$, and let two numbers $s, p \in \mathbb{R}$ s.t. $0 \leq s \leq 2$ and $1 \leq p \leq \infty$. Then $\forall \mathbf{u} \in [H^{s+1}(K)]^2$, there exists a function $u_\pi \in [\mathbb{P}_2(K)]^2$ s.t.:*

$$||\mathbf{u} - \mathbf{u}_\pi||_{L^p(K)} + h_K|\mathbf{u} - \mathbf{u}_\pi|_{W^{1,p}(K)} \leq Ch_K^{s+1}|\mathbf{u}|_{W^{s+1,p}(K)} \tag{4.51}$$

where C depends only on method order (here $k = 2$) and on γ defined in Assumption 3.

For what concerns trilinear form the following lemma holds (proof and derivation in Beirão da Veiga et al. [2018]).

Lemma 3 *Let $\hat{C}_h$ be the constant defined previously. Then it holds $\forall\, \mathbf{v}, \mathbf{z}, \mathbf{w} \in \mathbf{V}$ that*

$$|\tilde{c}_h(\mathbf{v}; \mathbf{v}, \mathbf{w}) - \tilde{c}_h(\mathbf{z}; \mathbf{z}, \mathbf{w})| \leq \hat{C}_h(||\mathbf{z}||_{\mathbf{V}}||\mathbf{w}||_{\mathbf{V}} + ||\mathbf{v} - \mathbf{z} + \mathbf{w}||_{\mathbf{V}}(||\mathbf{v}||_{\mathbf{V}} + ||\mathbf{z}||_{\mathbf{V}}))||\mathbf{w}||_{\mathbf{V}} \tag{4.52}$$

For discrete external force $\mathbf{f}_h$ the following lemma holds (Beirão da Veiga et al. [2018]):

Lemma 4 *Let $\mathbf{f}_h$ be defined as in (4.40) assuming $\mathbf{f} \in H^{s+1}(\Omega)$, $-1 \leq s \leq 2$. Then it is true that $\forall \mathbf{v}_h \in \mathbf{V}_h$:*

$$|\mathbf{f}_h - \mathbf{f}, \mathbf{v}_h| \leq Ch^{s+2}|\mathbf{f}|_{s+1}|\mathbf{v}_h|_{\mathbf{V}} \tag{4.53}$$

Moreover as observed in Beirão da Veiga et al. [2018] $\mathbf{Z}_h$ approximates $\mathbf{Z}$ with the same accuracy order as $\mathbf{V}_h$. Now convergence theorem in Beirão da Veiga et al. [2018] (for the proof see again Beirão da Veiga et al. [2018]) is reported.

Theorem 10 *Under assumption (4.6) for γ and under (4.46) for γ_h, let $\mathbf{u}$ be the solution of continuous problem (4.3) and $\mathbf{u}_h$ the solution for discrete problem (4.42). Assuming $\mathbf{u}, \mathbf{f} \in [H^{s+1}(\Omega)]^2$,$0 \leq s \leq 2$, it holds*

$$||\mathbf{u} - \mathbf{u}_h||_{\mathbf{V}} \leq h^s\mathcal{N}(\mathbf{u}; \nu, \gamma, \gamma_h) + h^{s+2}\mathcal{S}(\mathbf{f}; \nu, \gamma_h) \tag{4.54}$$

where $\mathcal{N}$ and $\mathcal{S}$ are function independent of h. Therefore the error, generally, scales as h^2.

Again, as in Stokes, error on velocity field does not depend on pressure.

For pressure it is possible to arrive to a similar result using inf-sup condition (see Beirão da Veiga et al. [2018]).

Theorem 11 *Let $(\mathbf{u}, p) \in \mathbf{V} \times Q$ be the solution of continuous problem (4.3) and $(\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h(Q_h^N)$ the solution of discrete problem (4.42). Then,*

$$||p - p_h||_Q \leq Ch^s|p|_s + Ch^{s+2}|\mathbf{f}|_{s+1} + h^s\mathcal{H}(\mathbf{u}; \nu, \gamma, \gamma_h) \tag{4.55}$$

where s has the same meaning as the analogue theorem for velocity and $\mathcal{H}$ is a function independent of h. Therefore error on pressure, generally, scales as h^2.

4.3 Newton's method

The Navier-Stokes equations are a nonlinear system of equations, therefore it is not possible to write the problem as an algebraic linear system and so to solve it like done for Stokes case. It is possible to use an iterative method in order to converge to solution of non linear problem.

Looking at work done in Gunzburger and Peterson [1991] here Newton's method is chosen. Given the discrete problem in (4.42) it is possible to write it in the equivalent form:

$$\begin{aligned} &\text{find} \quad \mathbf{u}_h \in \mathbf{V}_h \quad \text{and} \quad p_h \in Q_h\,(Q_h^N) \quad \text{such that} \\ &a_h(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) + c_h(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{u}_h, q) = (\mathbf{f}_h, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h, q \in Q_h(Q_h^N) \end{aligned} \tag{4.56}$$

Now Newton's method proceeds as described here. First of all an initial guess $\mathbf{u}_0 \in \mathbf{V}_h$ for velocity is chosen. It is important to observe that no initial guess of pressure is necessary. The iterates $\{\mathbf{u}_m \in \mathbf{V}_h, p_m \in Q^h\}_{m=1,2,\ldots}$ of Newton's method are defined solving:

$$\begin{aligned} &a_h(\mathbf{u}_m, \mathbf{v}_h) + c_h(\mathbf{u}_m, \mathbf{u}_{m-1}, \mathbf{v}_h) + c_h(\mathbf{u}_{m-1}, \mathbf{u}_m, \mathbf{v}_h) + b(\mathbf{v}_h, p_m) + b(\mathbf{u}_m, q) = \\ &(\mathbf{f}_h, \mathbf{v}_h) + c_h(\mathbf{u}_{m-1}; \mathbf{u}_{m-1}, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h, q \in Q_h\,(Q_h^N) \quad \text{and for} \quad m = 1,2,\ldots \end{aligned} \tag{4.57}$$

As reported in Gunzburger and Peterson [1991], if $(Re, \mathbf{u}(Re))$, where $Re = 1/\nu$ in the notation used here, is "a point of a nonsingular branch" and if initial estimate $\mathbf{u}_0$ is sufficiently near the solution $\mathbf{u}_h$ of discrete problem (4.42) the sequence of iterates of Newton's method $\{\mathbf{u}_m \in \mathbf{V}_h, p_m \in Q^h\}_{m=1,2,\ldots}$ converges *quadratically* to $(\mathbf{u}_h, p_h)$. Here as initial guess $\mathbf{u}_0$ the solution of the related (same data) Stokes problem $\mathbf{u}_S$ is chosen, solving it as described in Chapter 3.

4.4 Implementation: trilinear form and right-hand side

Here it is not explained how stiffness and pressure matrices as well as errors are implemented because their construction is the same as Stokes problem. The new parts are related to the trilinear form $c_h(\cdot; \cdot, \cdot)$ and to the right-hand side.

4.4.1 Trilinear form

From definition of local discrete trilinear form (4.34) it is clear that $\mathbf{\Pi}_1^{0,K}\nabla\mathbf{w}_h$ and $\mathbf{\Pi}_2^{0,K}\mathbf{v}_h$ must be computed for every $\mathbf{w}_h, \mathbf{v}_h \in \mathbf{V}_h$ given an element K. Computation for $\mathbf{\Pi}_1^{0,K}\nabla\mathbf{w}_h$ has already been described for gradient error on velocity in Stokes case, so it will not be repeated.

Now, to compute operator $\mathbf{\Pi}_2^{0,K} : \mathbf{V}_h \to [\mathbb{P}_2(K)]^2$, details about part of proof of Proposition 8 done in Vacca [2017] must be reported adapted to this case. From Vacca [2017], given $\mathbf{v}_h \in \mathbf{V}_h$ and $\mathbf{q}_2 \in [\mathbb{P}_2(K)]^2$ it is set:

$$\mathbf{q}_2 = \nabla q_3 + \mathbf{g}_2^{\perp} \tag{4.58}$$

where $q_3 \in \mathbb{P}_3(K)/\mathbb{R}$ (it is a polynomial of degree ≤ 3 but not a constant) and $\mathbf{g}_2^{\perp} \in \mathcal{G}_2^{\perp}(K)$ (space $\mathcal{G}_2^{\perp}(K)$ has already been defined in Stokes chapter). Now the following computation can be done (see Vacca [2017]):

$$\begin{gathered}\int_K \mathbf{v}_h \cdot \mathbf{q}_2 \, dK = \int_K \mathbf{v}_h \cdot (\nabla q_3 + \mathbf{g}_2^{\perp}) \, dK = \\ -\int_K \mathrm{div}\mathbf{v}_h \, q_3 \, dK + \int_K \mathbf{\Pi}_2^{\nabla,K}\mathbf{v}_h \cdot \mathbf{g}_2^{\perp} \, dK + \int_{\partial K} q_3 \mathbf{v}_h \cdot \mathbf{n} ds\end{gathered} \tag{4.59}$$

The last equality shows that the starting integral is computable. In fact first integral can be computed using $Dv3$, as already explained, the second one can be computed using operator $\mathbf{\Pi}_2^{\nabla,K}\mathbf{v}_h$ computed building bilinear form $a_h(\cdot,\cdot)$ as shown in Stokes case, the last one can be computed using $Dv1$ and $Dv2$.

Now it is defined the vector-valued monomial basis $\mathcal{M}_2^2(K)$ for $[\mathbb{P}_2(K)]^2$. Elements in $\mathcal{M}_2^2(K)$ are like the ones in (2.63) with monomials $m_i \in \mathbb{P}_2(K)$ in non-zero entries of vectors. Therefore $\dim(\mathcal{M}_2^2(K)) = 6 \cdot 2 = 12$. Using general definition (4.15) of $\mathbf{\Pi}_2^{0,K} : \mathbf{V}_h \to [\mathbb{P}_2(K)]^2$ and basis $\mathcal{M}_2^2(K)$ it is possible to write:

$$\int_K \mathbf{\Pi}_2^{0,K}\mathbf{v}_h \cdot \mathbf{q}_i = \int_K \mathbf{v}_h \cdot \mathbf{q}_i \quad \text{for} \quad i = 1, ..., 12, \quad \text{with} \quad \mathbf{q}_i \in \mathcal{M}_2^2(K) \tag{4.60}$$

$\mathbf{\Pi}_2^{0,K}\mathbf{v}_h \in [\mathbb{P}_2(K)]^2$ so:

$$\mathbf{\Pi}_2^{0,K}\mathbf{v}_h = \sum_{j=1}^{12} t^j \mathbf{q}_j \quad \text{where} \quad \mathbf{q}_j \in \mathcal{M}_2^2(K) \tag{4.61}$$

Combining (4.61) and (4.60) it holds

$$\sum_{j=1}^{12} t^j \int_K \mathbf{q}_j \cdot \mathbf{q}_i \, dK = \int_K \mathbf{v}_h \cdot \mathbf{q}_i \, dK \quad \text{for} \quad i = 1, ..., 12, \quad \text{with} \quad \mathbf{q}_i \in \mathcal{M}_2^2(K) \tag{4.62}$$

Let now the matrix $\underline{\mathbf{M}} \in \mathbb{R}^{12\times 12}$ and vector $\underline{\mathbf{l}} \in \mathbb{R}^{12\times 1}$ be such that:

$$\underline{\mathbf{M}}_{ij} = \int_K \mathbf{q}_j \cdot \mathbf{q}_i \, dK \qquad \underline{\mathbf{l}}_i = \int_K \mathbf{v}_h \cdot \mathbf{q}_i \, dK \tag{4.63}$$

In this way (4.62) can be rewritten as linear system:

$$\underline{\mathbf{M}}\mathbf{t} = \underline{\mathbf{l}} \tag{4.64}$$

where $\mathbf{t} \in \mathbb{R}^{12\times 1}$ such that $\mathbf{t}_j = t^j$.

$\underline{\mathbf{l}}_i$ values are computable from definition thanks to formula (4.59) which uses polynomial decomposition in (4.61). Therefore, in the following, some decompositions considering polygon K of elements of space $\mathcal{M}_2^2(K)$ using the same denomination as in (4.61) are done.

$$\mathbf{q}_2 = \begin{bmatrix}1\\0\end{bmatrix} \to q_3 = x - x_K \quad \mathbf{g}_2^\perp = \begin{bmatrix}0\\0\end{bmatrix} \tag{4.65}$$

$$\mathbf{q}_2 = \begin{bmatrix}\frac{x-x_K}{h_K}\\0\end{bmatrix} \to q_3 = \frac{(x-x_K)^2}{2h_K} \quad \mathbf{g}_2^\perp = \begin{bmatrix}0\\0\end{bmatrix} \tag{4.66}$$

$$\mathbf{q}_2 = \begin{bmatrix}\frac{y-y_K}{h_K}\\0\end{bmatrix} \to q_3 = 0 \quad \mathbf{g}_2^\perp = \begin{bmatrix}\frac{y-y_K}{h_K}\\0\end{bmatrix} \tag{4.67}$$

$$\mathbf{q}_2 = \begin{bmatrix}\frac{(x-x_K)^2}{h_K^2}\\0\end{bmatrix} \to q_3 = \frac{(x-x_K)^3}{3h_K} \quad \mathbf{g}_2^\perp = \begin{bmatrix}0\\0\end{bmatrix} \tag{4.68}$$

$$\mathbf{q}_2 = \begin{bmatrix}\frac{(y-y_K)^2}{h_K^2}\\0\end{bmatrix} \to q_3 = 0 \quad \mathbf{g}_2^\perp = \begin{bmatrix}\frac{(y-y_K)^2}{h_K^2}\\0\end{bmatrix} \tag{4.69}$$

$$\mathbf{q}_2 = \begin{bmatrix}\frac{(x-x_K)(y-y_K)}{h_K^2}\\0\end{bmatrix} \to q_3 = 0 \quad \mathbf{g}_2^\perp = \begin{bmatrix}\frac{(x-x_K)(y-y_K)}{h_K^2}\\0\end{bmatrix} \tag{4.70}$$

$$\mathbf{q}_2 = \begin{bmatrix}0\\1\end{bmatrix} \to q_3 = y - y_K \quad \mathbf{g}_2^\perp = \begin{bmatrix}0\\0\end{bmatrix} \tag{4.71}$$

$$\mathbf{q}_2 = \begin{bmatrix}0\\\frac{x-x_K}{h_K}\end{bmatrix} \to q_3 = 0 \quad \mathbf{g}_2^\perp = \begin{bmatrix}0\\\frac{x-x_K}{h_K}\end{bmatrix} \tag{4.72}$$

$$\mathbf{q}_2 = \begin{bmatrix}0\\\frac{y-y_K}{h_K}\end{bmatrix} \to q_3 = \frac{y-y_K}{h_K} \quad \mathbf{g}_2^\perp = \begin{bmatrix}0\\0\end{bmatrix} \tag{4.73}$$

$$\mathbf{q}_2 = \begin{bmatrix}0\\\frac{(x-x_K)^2}{h_K^2}\end{bmatrix} \to q_3 = 0 \quad \mathbf{g}_2^\perp = \begin{bmatrix}0\\\frac{(x-x_K)^2}{h_K^2}\end{bmatrix} \tag{4.74}$$

$$\mathbf{q}_2 = \begin{bmatrix}0\\\frac{(y-y_K)^2}{h_K^2}\end{bmatrix} \to q_3 = \frac{(y-y_K)^3}{3h_K^2} \quad \mathbf{g}_2^\perp = \begin{bmatrix}0\\0\end{bmatrix} \tag{4.75}$$

$$\mathbf{q}_2 = \begin{bmatrix}0\\\frac{(x-x_K)(y-y_K)}{h_K^2}\end{bmatrix} \to q_3 = 0 \quad \mathbf{g}_2^\perp = \begin{bmatrix}0\\\frac{(x-x_K)(y-y_K)}{h_K^2}\end{bmatrix} \tag{4.76}$$

Recalling definition (4.34) of $c_h^K(\cdot;\cdot,\cdot)$ let now tridimensional matrix $\mathbf{Z} \in \mathbb{R}^{12\times 12\times 12}$ be defined as:

$$\begin{aligned} \mathbf{Z}_{abc} = \int_K (\mathbf{Q}_a\,\mathbf{q}_b)\cdot \mathbf{q}_c\,dK \quad &\text{for} \quad a,b,c = 1,...,12 \\ \text{with} \quad \mathbf{Q}_a \in \mathcal{M}_1^{2\times 2}(K) \quad &\text{and} \quad \mathbf{q}_b, \mathbf{q}_c \in \mathcal{M}_2^2(K) \end{aligned} \tag{4.77}$$

In other words matrix $\mathbf{Z}$ contains the values of integral used in $c_h^K(\cdot;\cdot,\cdot)$ definition calculated with every combination of one element of $\mathcal{M}_1^{2\times 2}(K)$ and two elements of $\mathcal{M}_2^2(K)$. Therefore, given $\mathbf{w}_h, \mathbf{u}_h, \mathbf{v}_h \in \mathbf{V}_h^K$, in order to compute $c_h^K(\mathbf{w}_h;\mathbf{u}_h,\mathbf{v}_h)$, components of $\mathbf{\Pi}_1^{0,K}\nabla\mathbf{w}_h$ respect to basis $\mathcal{M}_1^{2\times 2}(K)$ are obtained, using previous explanation, and saved in a vector as well as component of $\mathbf{\Pi}_2^{0,K}\mathbf{u}_h$ and $\mathbf{\Pi}_2^{0,K}\mathbf{v}_h$ respect to basis $\mathcal{M}_2^2$. Then every entry of matrix $\mathbf{Z}$ is multiplied by the product of the three components of $\mathbf{\Pi}_1^{0,K}\nabla\mathbf{w}_h$, $\mathbf{\Pi}_2^{0,K}\mathbf{u}_h$ and $\mathbf{\Pi}_2^{0,K}\mathbf{v}_h$, corresponding to basis elements used to compute that entry of $\mathbf{Z}$. Finally all values are summed obtaining $c_h^K(\mathbf{w}_h;\mathbf{u}_h,\mathbf{v}_h)$.

4.4.2 Right-hand side

Given an element $K \in \mathcal{T}_h$ and the related local canonical basis functions $\{\boldsymbol{\varphi}_i\}_{i=1,...,N^K}$ of $\mathbf{V}_h^K$, right-hand side $(\mathbf{f}_h, \boldsymbol{\varphi}_i)$ is given by formula (4.41):

$$(\mathbf{f}_h, \boldsymbol{\varphi}_i) = \int_K \mathbf{f}\cdot\mathbf{\Pi}_2^{0,K}\boldsymbol{\varphi}_i\,dK \tag{4.78}$$

$\mathbf{\Pi}_2^{0,K}\boldsymbol{\varphi}_i$ is obtained using the explanation above to compute L^2-projection operator. Then scalar product between $\mathbf{f}$ and degree 2 polynomial is done. In this way the integral can be approximate using a bidimensional Simpson quadrature rule. In fact Simpson formula is of fourth order (error scales as h^4) so it should be compatible with how error of VEM Navier-Stokes problem scales.

4.4.3 Linearized equations system

Now considering Newton iteration (4.57) where $\mathbf{u}_{m-1}$ is fixed, it is possible to write from (4.57) a linear system. Considering global canonical basis functions of $\mathbf{V}_h$ $\{\boldsymbol{\varphi}_i\}_{i=1,...,N_{dofs}}$ and of Q_h (Q_h^N) $\{q_j\}_{j=1,...,Q_{dofs}(+1)}$, (4.57) can be rewritten as:

$$\begin{cases} a_h(\mathbf{u}_m,\boldsymbol{\varphi}_i) + c_h(\mathbf{u}_m,\mathbf{u}_{m-1},\boldsymbol{\varphi}_i) + c_h(\mathbf{u}_{m-1},\mathbf{u}_m,\boldsymbol{\varphi}_i) + b(\boldsymbol{\varphi}_i,p_m) = \\ \qquad\qquad (\mathbf{f}_h,\boldsymbol{\varphi}_i) + c_h(\mathbf{u}_{m-1};\mathbf{u}_{m-1},\boldsymbol{\varphi}_i) \quad \text{for} \quad i = 1,...,N_{dofs} \\ b(\mathbf{u}_m,q_j) = 0 \quad \text{for} \quad j = 1,...,Q_{dofs}(+1) \end{cases} \tag{4.79}$$

Considering that unknown velocity $\mathbf{u}_m \in \mathbf{V}_h$ and pressure $p_m \in Q_h$ (Q_h^N), it is possible to expand them respect to canonical basis.

$$\mathbf{u}_m = \sum_{z=1}^{N_{dofs}} u_m^z\boldsymbol{\varphi}_z \tag{4.80}$$

$$p_m = \sum_{z=1}^{Q_{dofs}} p_m^z q_z \tag{4.81}$$

Therefore (4.79) becomes:

$$\begin{cases} \displaystyle\sum_{z=1}^{N_{dofs}} u_m^z \, a_h(\boldsymbol{\varphi}_z, \boldsymbol{\varphi}_i) + \sum_{z=1}^{N_{dofs}} u_m^z c_h(\boldsymbol{\varphi}_z, \mathbf{u}_{m-1}, \boldsymbol{\varphi}_i) + \sum_{z=1}^{N_{dofs}} u_m^z c_h(\mathbf{u}_{m-1}, \boldsymbol{\varphi}_z, \boldsymbol{\varphi}_i) \\ \quad \displaystyle + \sum_{z=1}^{Q_{dofs}} b(\boldsymbol{\varphi}_i, q_z) = (\mathbf{f}_h, \boldsymbol{\varphi}_i) + c_h(\mathbf{u}_{m-1}; \mathbf{u}_{m-1}, \boldsymbol{\varphi}_i) \quad \text{for} \quad i = 1, ..., N_{dofs} \\ \\ \displaystyle\sum_{z=1}^{N_{dofs}} u_m^z b(\boldsymbol{\varphi}_z, q_j) = 0 \quad \text{for} \quad j = 1, ..., Q_{dofs}(+1) \end{cases} \tag{4.82}$$

Here the derivation is done with the limitation of homogeneous boundary conditions on Dirichlet boundary for simplicity and without the constrain $\int_\Omega p_h \, d\Omega = 0$.

Then, it is natural to define the global stiffness matrix $\mathbf{A} \in \mathbb{R}^{N_{dofs} \times N_{dofs}}$ and global pressure matrix $\mathbf{P} \in \mathbb{R}^{N_{dofs} \times Q_{dofs}}$:

$$\mathbf{A}_{iz} = a_h(\boldsymbol{\varphi}_z, \boldsymbol{\varphi}_i) \tag{4.83}$$

$$\mathbf{P}_{zj} = b(\boldsymbol{\varphi}_z, q_j) \tag{4.84}$$

Those matrices are obtained assembling the local ones as done for Stokes cases. Moreover it is possible to define the global vector $\underline{\mathbf{f}} \in \mathbb{R}^{N_{dofs}}$:

$$\underline{\mathbf{f}}_i = (\mathbf{f}_h, \boldsymbol{\varphi}_i) \tag{4.85}$$

assembling the local vectors computed using explanation in previous part.

For what concerns parts with c let now be the vector $\mathbf{c}^s \in \mathbb{R}^{N_{dofs}}$:

$$\mathbf{c}_i^s = c_h(\mathbf{u}_{m-1}; \mathbf{u}_{m-1}, \boldsymbol{\varphi}_i) \tag{4.86}$$

Moreover matrices $\mathbf{C}^1 \in \mathbb{R}^{N_{dofs} \times N_{dofs}}$ and $\mathbf{C}^2 \in \mathbb{R}^{N_{dofs} \times N_{dofs}}$ are defines as

$$\mathbf{C}_{iz}^1 = c_h(\boldsymbol{\varphi}_z; \mathbf{u}_{m-1}, \boldsymbol{\varphi}_i) \tag{4.87}$$

$$\mathbf{C}_{iz}^2 = c_h(\mathbf{u}_{m-1}; \boldsymbol{\varphi}_z, \boldsymbol{\varphi}_i) = \tag{4.88}$$

$\mathbf{c}^s$,$\mathbf{C}^1$ and $\mathbf{C}^2$ are obtained assembling the local vector/matrices $\mathbf{c}^{s,K}$,$\mathbf{C}^{s,K}$ and $\mathbf{C}^{s,K}$.

Therefore the linear system related to one iteration of Newton's method is:

$$\begin{cases} \mathbf{A}\underline{\mathbf{u}_m} + \mathbf{C}^1\underline{\mathbf{u}_m} + \mathbf{C}^2\underline{\mathbf{u}_m} + \mathbf{P}\underline{p_m} = \underline{\mathbf{f}} + \mathbf{c}^s \\ \mathbf{P}^T\underline{\mathbf{u}_m} = \mathbf{0} \end{cases} \tag{4.89}$$

where $\underline{\mathbf{u}_m}$ and $\underline{p_m}$ are vectors containing components, respectively, of $\mathbf{u}_m$ and p_m respect to canonical basis. If there are only Dirichlet boundary conditions it is necessary to impose on the domain $\int_\Omega p_h \, d\Omega = 0$. This change the parts of system related to $\mathbf{P}$ and $\underline{p_m}$ in the same way as described for Stokes case. Furthermore if non-homogeneous Dirichlet boundary conditions are present right-hand sides of system (4.89) must be modified similarly to what was done for Stokes case. (4.89), in matrix form appears as:

$$\begin{bmatrix} \mathbf{A} + \mathbf{C}^1 + \mathbf{C}^2 & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}_m} \\ \underline{p_m} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{f}} + \mathbf{c}^s \\ \mathbf{0} \end{bmatrix} \tag{4.90}$$

4.4.4 Other boundary conditions

Accordingly to the work in Breuer et al. [2000] the condition at the outflow are given by:

$$\frac{\partial \mathbf{u}}{\partial x} = \mathbf{0} \tag{4.91}$$

Remembering that Neumann conditions on the outflow boundary correspond to:

$$\begin{bmatrix} \nu \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial x} \end{bmatrix} = \begin{bmatrix} -p \\ 0 \end{bmatrix} \tag{4.92}$$

the new condition can be obtained adding at the right hand side the pressure dependent term:

$$-\int_{\partial\Omega_N} p \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \mathbf{v} \tag{4.93}$$

where $\mathbf{v} \in \mathbf{V}$ is a test function. In the discrete formulation of the problem an analogous term will be added. Writing the previous integral on the left-hand side it will change its sign. Computationally this leads to a new matrix term multiplying $\underline{p_m}$. Let now be the matrix $\mathbf{H} \in \mathbb{R}^{N_{dofs} \times Q_{dofs}+1}$ where rows are related to global canonical basis functions of space $\mathbf{V}_h$ and columns to global canonical basis functions of space $\mathbf{Q}_h^N$. So the integral related to each row to be computed is, using $\boldsymbol{\varphi}_i$ to indicate canonical basis functions of space $\mathbf{V}_h$,

$$\int_{\partial\Omega_N} p \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \boldsymbol{\varphi}_i \tag{4.94}$$

Expanding p_h using canonical basis of Q_h^N is it possible to obtain the new term in matrix form:

$$\mathbf{H}\underline{p_m} \tag{4.95}$$

where:

$$\mathbf{H}_{ij} = \int_{\partial\Omega_N} q_j \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \boldsymbol{\varphi}_i \tag{4.96}$$

where the only q_j that will give a non-zero contribution are those related to elements that have at least one edge of outflow boundary. $\boldsymbol{\varphi}_i$ will give non zero contribution if they are related to point or edges on the outflow boundary. q_j functions related to zero-order momentum and x-momentum will be constant because the outflow boundary is vertical so

x is constant. The whole integral will be split in the different contributions coming from each element near outflow boundary. Those integrals will be calculated with a Simpson formula. So the matrix in (4.90) will be modified as:

$$\begin{bmatrix} \mathbf{A} + \mathbf{C}^1 + \mathbf{C}^2 & \mathbf{P} + \mathbf{H} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \tag{4.97}$$

while the rest remains unchanged from Neumann conditions case.

4.5 Numerical tests and experiments

Before dealing with fluid dynamics test case a mathematical test case is taken into account in order to observe the behaviour of the method. All observations about errors plots and function plots are the same as Stokes case. In all cases 4 iterations of Newton's method are used.

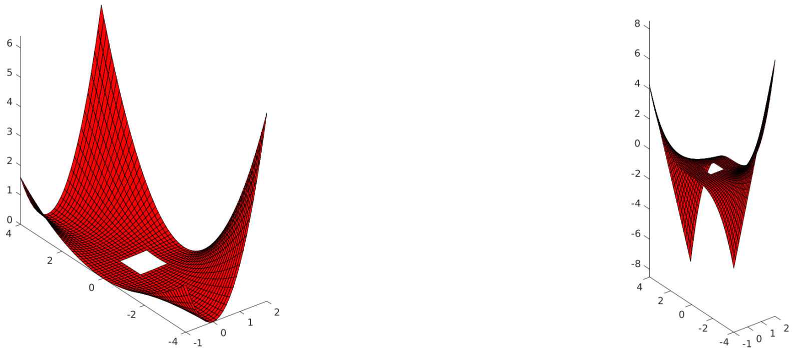
4.5.1 Mathematical test

Case 1

For the case considered $\mathbf{f}$ is chosen such that the solution is:

$$\mathbf{u} = \frac{1}{10} \begin{bmatrix} x^2y^2 \\ -2xy^3/3 \end{bmatrix} \qquad p = y^3 \tag{4.98}$$

Velocity field fulfills incompressibility condition. Non-homogeneous boundary condition equal to $\mathbf{u}$ are imposed on every borders. It is possible to see that pressure $\int_\Omega p \, d\Omega = 0$. Moreover $Re = 1$, so $\nu = 1$. The starting mesh used is shown in Figure 3.1 and 2 uniform refinements are done. Figure 4.1 shows plots of velocity components obtained with the most refined mesh. Errors plot is shown in Figure 4.2. It is possible to see that error on the pressure is dominant and that the two errors follow theoretical trend.



(a) u_x

(b) u_y

Figure 4.1: Plots of VEM discrete velocity solution components (case *1* Navier Stokes)

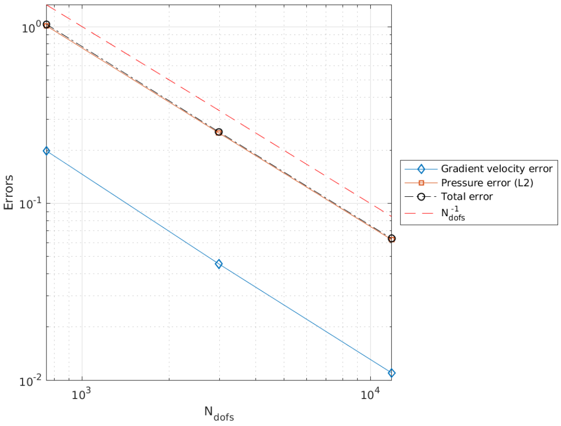


Figure 4.2: Plot of errors (case *1* Navier Stokes)

4.5.2 Fluid dynamics test case

Data are the same as the corresponding case with Stokes problem. Here ν means $1/Re$. As it will be shown, limiting the range to $Re < 60$, it is not necessary to use a stabilization method like SUPG (Beirão da Veiga et al. [2020]) with the adopted meshes. Analysis for different Reynolds number is done. Following Breuer et al. [2000] conditions (4.91) are imposed at the outflow boundary. For other boundaries, conditions explained in introduction are adopted. In particular inflow condition is given by (3.142). In fact the fluid dinamics problem is adimensionalized choosing as characteristic velocity the maximum value u_{max} of parabolic function at inflow. Therefore in the adimensional case this maximum value will be equal to one. The characteristic length is given by the cylinder side in fact its length is equal to one in the domain. The external force is $\mathbf{f} = \mathbf{0}$. For all cases plots with velocity vectors corresponding to mesh nodes and with streamlines (blue lines in every plot) are shown.

Re=1

The case $Re = 1$ is solved on the mesh in Figure 4.3. The grid has one level of refinement and, so, there are elements with two different characteristic dimensions. Generally squares have a characteristic dimension of $h = 1/8$, while around the cylinder elements with $h = 1/16$ are adopted since here most interesting fluid dynamics effects can occur (boundary layer around cylinder, separation of boundary layer and potential eddies). In Figure 4.4 flow can be viewed, in particular Figure 4.4b shows the velocity field around the cylinder. It is possible to notice that situation is similar to Stokes one as the Reynolds number is still low. The plots show that there is no boundary layer separation with $Re = 1$ as written in Breuer et al. [2000].

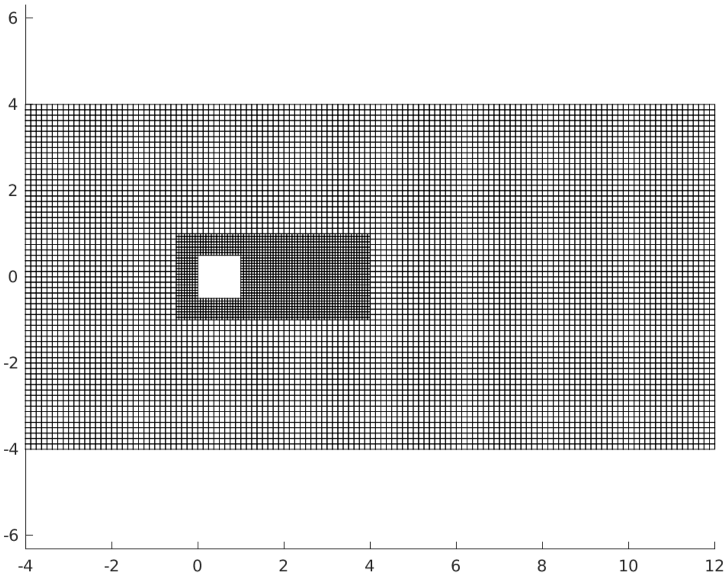
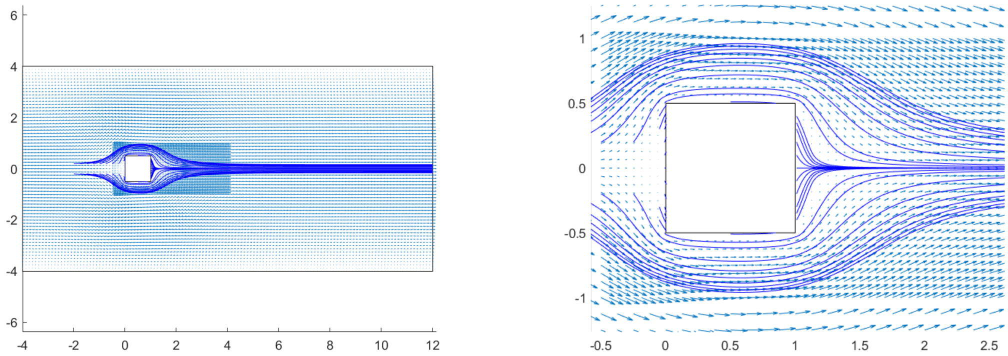


Figure 4.3: Mesh with one level of refinement used in Navier Stokes cases



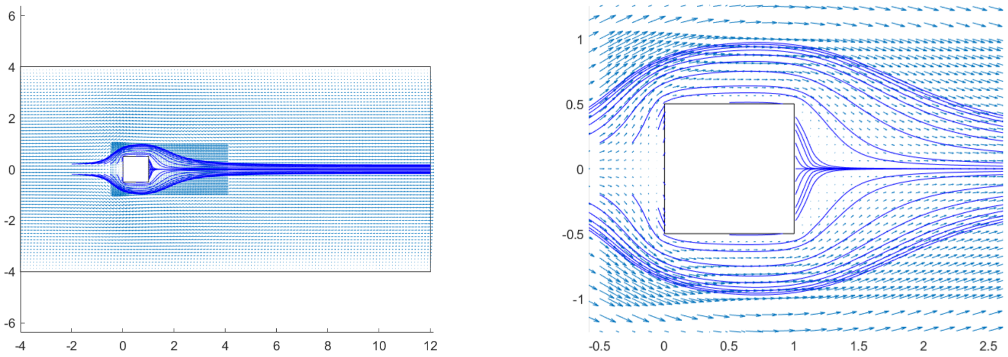
(a) Flow on the whole mesh

(b) Flow around cylinder

Figure 4.4: *Fluid dynamics test case* Navier-Stokes solution velocity field for $Re = 1$

Re=3,4,5

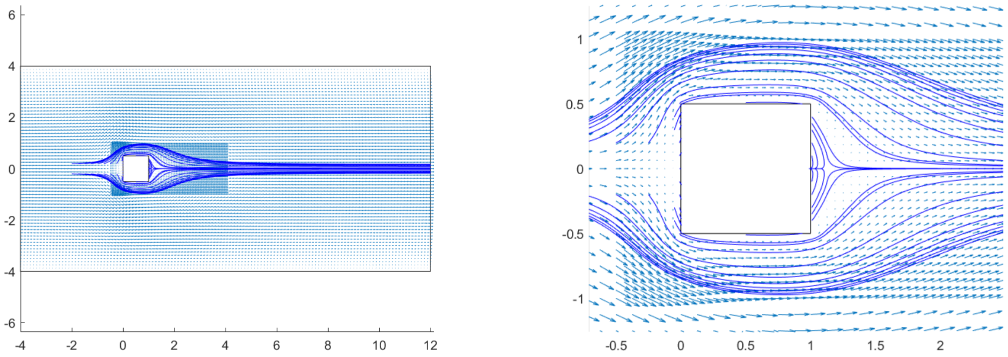
In this part situations with low Reynolds number (3,4 and 5) are investigated in order to understand when the separation of boundary layer starts, remembering that the corresponding Reynolds number Re_{crit} is estimated below $Re = 5$ in Breuer et al. [2000]. Above this value recirculation regions can be seen accordingly to Breuer et al. [2000], so flow around cylinder are monitored to observe when those regions occur. This means that



(a) Flow on the whole mesh

(b) Flow around cylinder

Figure 4.5: *Fluid dynamics test case* Navier-Stokes solution velocity field for $Re = 3$

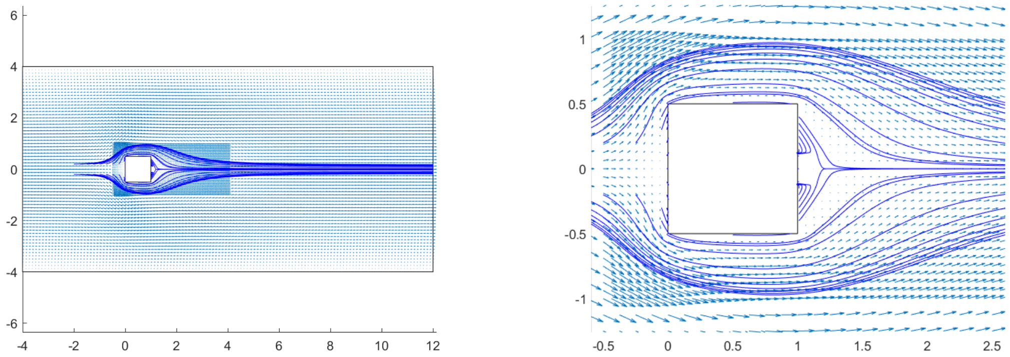


(a) Flow on the whole mesh

(b) Flow around cylinder

Figure 4.6: *Fluid dynamics test case* Navier-Stokes solution velocity field for $Re = 4$

some values of velocity component u_x are negative after the cylinder. Results are again obtained using the mesh in Figure 4.3. Figures 4.5,4.6 and 4.7 show the flows obtained, respectively, with $Re = 3$, $Re = 4$ and $Re = 5$. Observing the flows near the cylinder it is possible to notice that there is no separation of boundary layer at $Re = 3$, while for $Re = 5$ it is possible to see two small symmetric recircurlation regions which indicate that separation has started. For $Re = 4$ near the cylinder some values of u_x are negative, in fact small recirculation regions near cylinder wall can be seen. So simulations suggest that $Re_{crit} < 5$ like in Breuer et al. [2000]. Moreover $Re_{crit} > 3$ and, observing results for $Re = 4$, Re_{crit} is probably near $Re = 4$ with $Re_{crit} < 4$.



(a) Flow on the whole mesh

(b) Flow around cylinder

Figure 4.7: *Fluid dynamics test case* Navier-Stokes solution velocity field for $Re = 5$

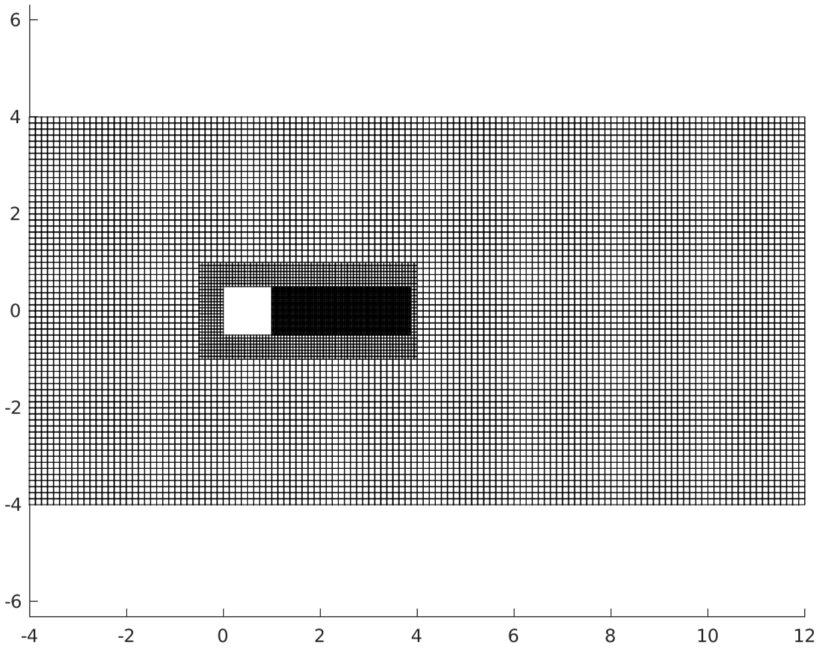
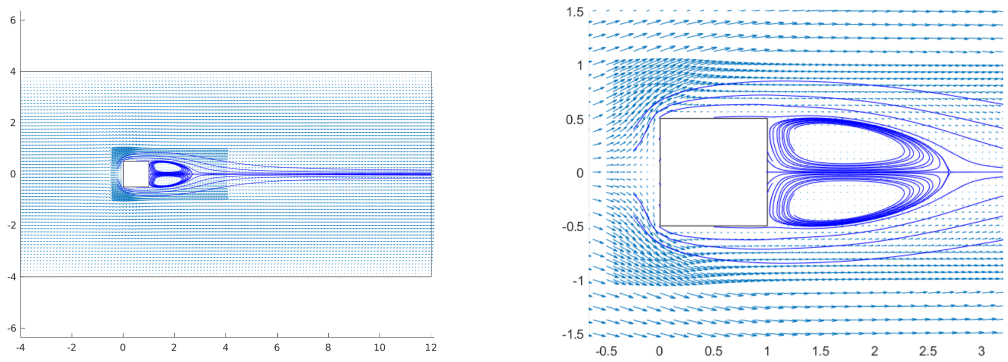


Figure 4.8: Mesh with two levels of refinement for the *Fluid dynamics test case*

Re=30

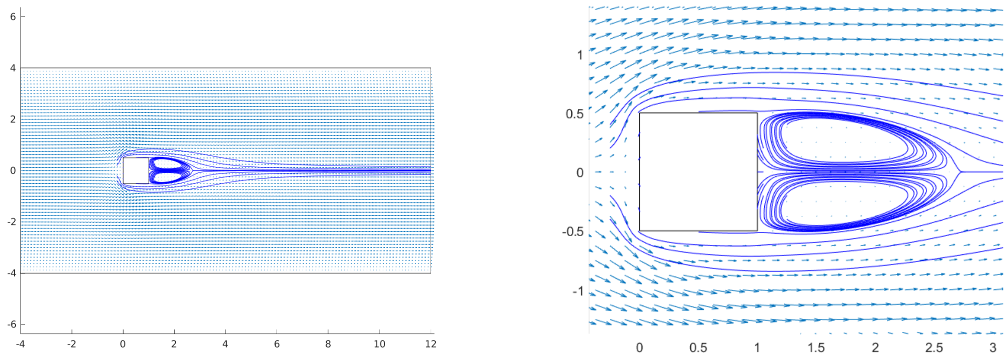
The case corresponding to $Re = 30$ is particularly important because velocity field plots results can be directly compared with a figure of Breuer et al. [2000]. Figure 4.9 shows velocity field with streamlines obtained using mesh in Figure 4.3. Furthermore the same



(a) Flow on the whole mesh

(b) Flow around cylinder

Figure 4.9: *Fluid dynamics test case* Navier-Stokes solution velocity field for $Re = 30$ (mesh with one level of refinement)



(a) Flow on the whole mesh

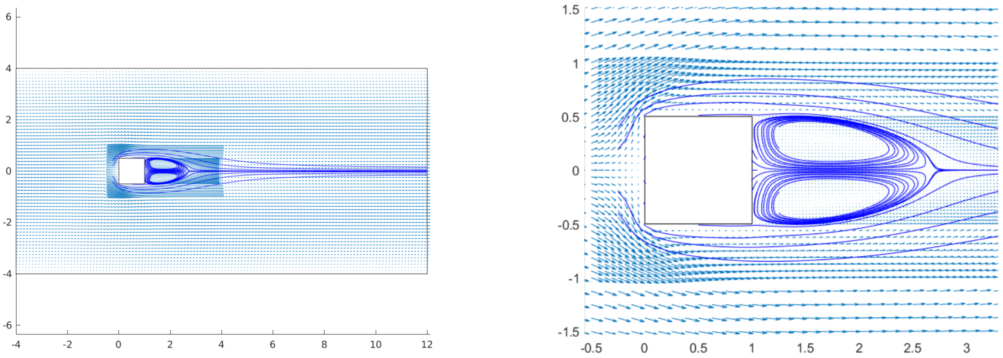
(b) Flow around cylinder

Figure 4.10: *Fluid dynamics test case* Navier-Stokes solution velocity field for $Re = 30$ (mesh with no level of refinement)

problem is solved with a mesh without hanging nodes (all elements have characteristic dimension $h = 1/8$) and a new mesh obtained adding a level of refinement to mesh in Figure 4.3 in the wake region right after cylinder. In this last case new elements will have a characteristic dimension of $h = 1/32$. The mesh is shown in Figure 4.8. Figure 4.10 and Figure 4.11 show, respectively, the flow obtained using a mesh with no level of refinement (no hanging nodes) and with two levels of refinement (see Figure 4.8). It is possible to see that solution shows well the vortices with the mesh without hanging nodes ($h = 1/8$ for every element), even if in that case the mesh Reynolds number Re_m defined as,

$$Re_m = Re \cdot h, \tag{4.99}$$

is $Re_m = 30/8 = 3.75 > 1$.



(a) Flow on the whole mesh

(b) Flow around cylinder

Figure 4.11: *Fluid dynamics test case* Navier-Stokes solution velocity field for $Re = 30$ (mesh with two levels of refinement)

It is possible to notice that two symmetric eddies compare in the wake like in Breuer et al. [2000]. Their length is about 1.7 in adimensional units similarly to Breuer et al. [2000]. Therefore computational fluid dynamics literature confirms results shown here.

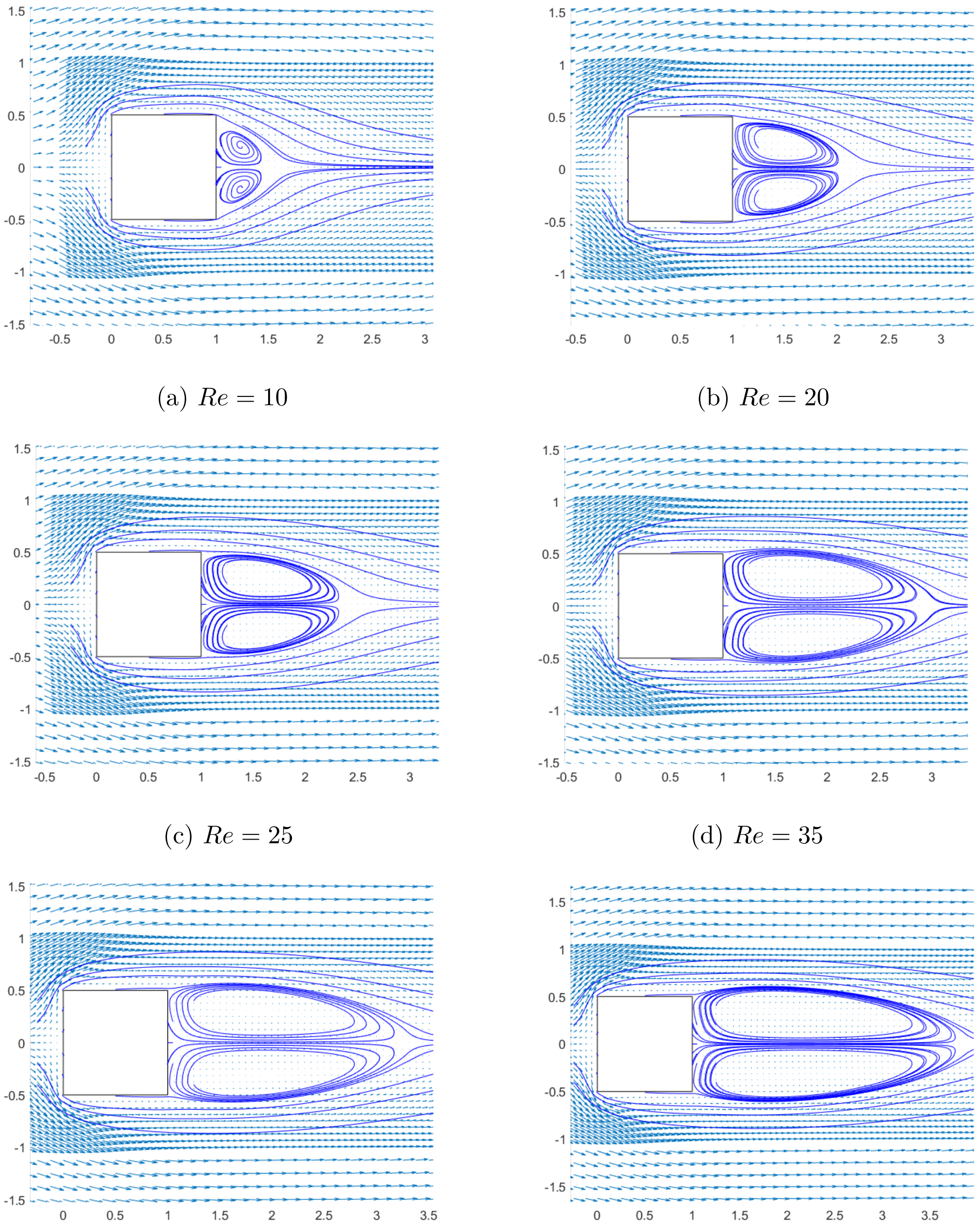
Different Reynolds number ($Re < 60$) and Recirculation length

In order to understand the behaviour of eddies with respect to different values of Re, various simulations, using the mesh in Figure 4.3, are realized. Simulations start from $Re = 10$ up to $Re = 55$. Figure 4.12 shows velocity field for some values of the Reynolds number. It is easy to see that length of eddies grows increasing Reynolds number, as observed in Breuer et al. [2000]. It is interesting to analyse the trend of recirculation length, namely the length of vortices, with respect to the Reynolds number. In order to measure that length, here, the values of u_x on the longitudinal axis ($y = 0$) is considered. Recirculation length (L_r) is given by the difference between the (approximated) x-value where u_x changes its sign on longitudinal axis (x_{u_x}) and the abscissa $x_{square} = 1$ of the right side of the cylinder ($L_r = x_{u_x} - x_{square}$). Table 4.1 reports the values of recirculation length in different cases.

Analyzing the values obtained it is possible to empirically derived a linear relationship between recirculation length L_r and Reynold number Re. It is possible to write the following formula interpolating the data (see Figure 4.13):

$$L_r = 0.06Re - 0.1 \quad 10 \leq Re \leq 55 \tag{4.100}$$

Also in Breuer et al. [2000] a linear relation is observed, so results obtained make sense. In that case $L_r = 0.0554Re - 0.065$, so coefficients found here (see (4.100)) are an approximation of the ones in Breuer et al. [2000].



(a) $Re = 10$

(b) $Re = 20$

(c) $Re = 25$

(d) $Re = 35$

(e) $Re = 40$

(f) $Re = 50$

Figure 4.12: *Fluid dynamics test case* solution velocity field around cylinder for different Reynolds numbers (Re)

Re	L_r
10	0.5
15	0.8
20	1.1
25	1.4
30	1.7
35	2.0
40	2.3
45	2.6
50	2.9
55	3.2

Table 4.1: Values of recirculation length (L_r) for different Reynolds numbers (Re)

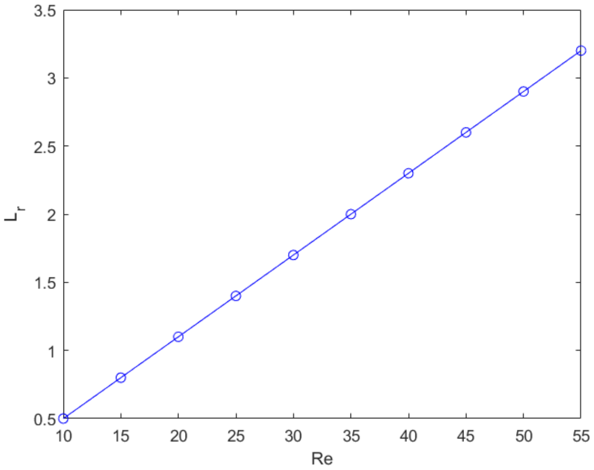


Figure 4.13: (Plot) Recirculation lengths (L_r) for different Reynolds numbers (Re)

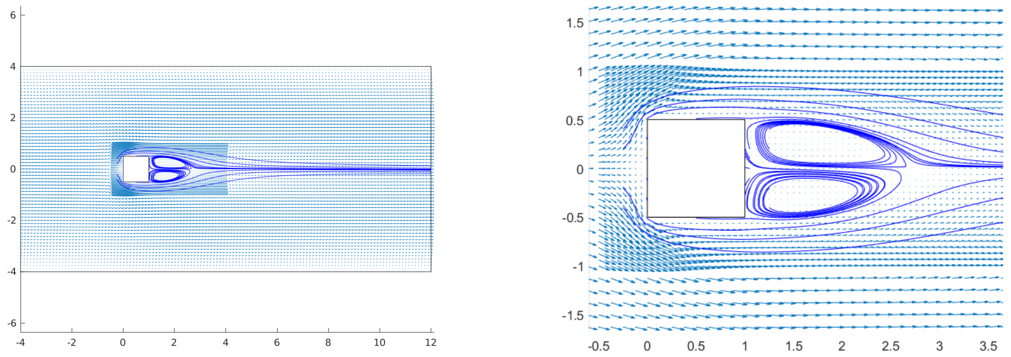
Oblique flow cases

Previous cases always use the parabolic inflow condition which is symmetric with respect to the longitudinal axis ($y = 0$). To explore what happens when a non-symmetric inflow condition is adopted the inflow vector $\mathbf{u}_{inflow}$ is rotated. The resulting boundary conditions are obtained multiplying the normal inflow vector function by the rotation matrix. Therefore

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} -\frac{1}{16}(y-4)(y+4) \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{16}(y-4)(y+4)\cos(\alpha) \\ -\frac{1}{16}(y-4)(y+4)\sin(\alpha) \end{bmatrix} \tag{4.101}$$

where α is the rotation angle in counterclockwise order with respect to axis $x = -4$. So, now, the y component of inflow condition is no more equal to zero.

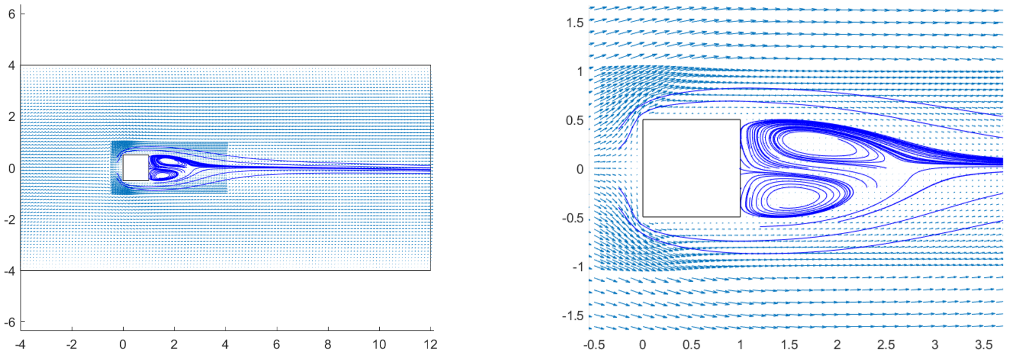
Figure 4.14 shows the velocity field obtained with $\alpha = 5$ and $Re = 30$. It is possible to see that the eddies, which appears also in the case without rotation, becomes slightly



(a) Flow on the whole mesh

(b) Flow around cylinder

Figure 4.14: *Fluid dynamics test case* Navier-Stokes solution velocity field for $Re = 30$ and rotation angle $\alpha = 5$



(a) Flow on the whole mesh

(b) Flow around cylinder

Figure 4.15: *Fluid dynamics test case* Navier-Stokes solution velocity field for $Re = 30$ and rotation angle $\alpha = 15$

asymmetric. Considering the same case with $\alpha = 15$, so a more rotated parabolic function, results shown in Figure 4.15 are obtained. It is possible to see that in this case the vortices are more asymmetric than $\alpha = 5$ case as expected due to the more rotated function at the inflow.

Chapter 5

Conclusions

In this last chapter results of the thesis are summarized.

First of all, adaptive mesh refinement for the cases considered in Chapter 2 confirms the fact that stabilization part is not necessary in the a posteriori estimator related to the diffusion problem as proved in Beirão da Veiga et al. [2021b].

Chapter 3 related to Stokes problem shows the application of a posteriori estimator in Wang et al. [2020] to some numerical tests. Again, as for the diffusion case, stabilization contribution has not the main importance in all the cases considered, however we do not know a theorem as for the diffusion case. Moreover adaptive mesh refinement in the fluid dynamics situation shows that the estimate of the errors is concentrated near the vertices of cylinder. Therefore the process builds a mesh adapted to the problem after some refinements steps. The flow obtained in this case without convection is similar to the corresponding case with Navier-Stokes $Re = 1$.

Finally Chapter 4 related to Navier-Stokes problem shows how VEM produces good results in the fluid dynamics situation. For $Re = 1$ there is no boundary layer separation. Increasing Reynolds number we found that eddies formation and, so, boundary layer separation starts at $Re < 5$ confirming results in Breuer et al. [2000]. For $Re = 30$ the flow, especially the vortices past the cylinder, appears as in Breuer et al. [2000] confirming that VEM gives a good solution for this physical problem. Increasing the Reynolds number we found as in Breuer et al. [2000] that recicurlation length (i.e. the length of vortices) increases as well. Analysing flows obtained with various Reynolds number a table with recirculation lengths and Reynolds number was built. In this way a linear relation was observed confirming, again, what found in Breuer et al. [2000].

Bibliography

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(01): 199–214, 2013a. doi: 10.1142/S0218202512500492. URL https://doi.org/10.1142/S0218202512500492.

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, 2013b. doi: 10.1137/120874746. URL https://doi.org/10.1137/120874746.

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(08):1541–1573, 2014.

L. Beirão da Veiga, C. Lovadina, and G. Vacca. Divergence free virtual elements for the Stokes problem on polygonal meshes, 2015. URL https://arxiv.org/abs/1510.01655v1.

L. Beirão da Veiga, C. Lovadina, and G. Vacca. Divergence free virtual elements for the Stokes problem on polygonal meshes. *ESAIM: M2AN*, 51(2):509–535, 2017. doi: 10.1051/m2an/2016032. URL https://doi.org/10.1051/m2an/2016032.

L. Beirão da Veiga, C. Lovadina, and G. Vacca. Virtual elements for the Navier–Stokes problem on polygonal meshes. *SIAM Journal on Numerical Analysis*, 56(3):1210–1242, 2018. doi: 10.1137/17M1132811. URL https://doi.org/10.1137/17M1132811.

L. Beirão da Veiga, F. Dassi, C. Lovadina, and G. Vacca. Supg-stabilized virtual elements for diffusion-convection problems: a robustness analysis, 2020. URL https://arxiv.org/abs/2012.01104v1.

L. Beirão da Veiga, C. Canuto, R. H. Nochetto, and G. Vacca. Equilibrium analysis of an immersed rigid leaflet by the virtual element method. *Math. Models Methods Appl. Sci.*, 31(7):1323–1372, 2021a.

L. Beirão da Veiga, C. Canuto, R. H. Nochetto, G. Vacca, and M. Verani. Adaptive VEM: stabilization-free a posteriori error analysis, 2021b. URL https://arxiv.org/abs/2111.07656.

D. Boffi, F. Brezzi, and M. Fortin. *Mixed Finite Element Methods and Applications*, volume 44 of *Springer Series in Computational Mathematics*. Springer, 2013.

S. C. Brenner and R. L. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, 2008.

M. Breuer, J. Bernsdorf, T. Zeiser, and F. Durst. Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume. *International Journal of Heat and Fluid Flow*, 21(2):186–196, 2000. ISSN 0142-727X. URL https://doi.org/10.1016/S0142-727X(99)00081-8.

A. Cangiani, E.H. Georgoulis, T. Pryer, and O. J. Sutton. A posteriori error estimates for the virtual element method. *Numer. Math.*, 137:857–893, 2017. doi: 10.1007/s00211-017-0891-9. URL https://doi.org/10.1007/s00211-017-0891-9.

Max D. Gunzburger and Janet S. Peterson. Predictor and steplength selection in continuation methods for the Navier-Stokes equations. *Computers Math. Applic.*, 22(8):73–81, 1991.

MathWorks. Matlab documentation center. URL http://www.mathworks.it/it/help/matlab/.

Giovanni Monegato. *Metodi e algoritmi per il calcolo numerico*. CLUT, 2008.

Stéphane Popinet. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics*, 190(2):572–600, 2003. ISSN 0021-9991. doi: https://doi.org/10.1016/S0021-9991(03)00298-5. URL https://www.sciencedirect.com/science/article/pii/S0021999103002985.

Stéphane Popinet. A quadtree-adaptive multigrid solver for the Serre–Green–Naghdi equations. *Journal of Computational Physics*, 302:336–358, 2015. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2015.09.009. URL https://www.sciencedirect.com/science/article/pii/S0021999115005902.

Alfio Quarteroni. *Modellistica Numerica per Problemi Differenziali*. Springer, 2016.

Oliver J. Sutton. The virtual element method in 50 lines of MATLAB. *Numer Algor*, 75:1141–1159, 2017. doi: 10.1007/s11075-016-0235-3. URL https://doi.org/10.1007/s11075-016-0235-3.

Giuseppe Vacca. An H^1-conforming virtual element method for Darcy equations and Brinkman equations. 2017. URL https://arxiv.org/abs/1701.07680v2.

G. Wang, Y. Wang, and Y. He. A posteriori error estimates for the virtual element method for the Stokes problem. *Journal of Scientific Computing*, 84(37), 2020. doi: 10.1007/s10915-020-01281-2. URL https://doi.org/10.1007/s10915-020-01281-2.