Development of a multiscale finite element method for incompressible flows in heterogeneous media

Qingqing Feng

Qingqing Feng. Development of a multiscale finite element method for incompressible flows in heterogeneous media. Numerical Analysis [cs.NA]. Université Paris Saclay (COmUE), 2019. English. NNT : 2019SACLX047 . tel-02325512

HAL Id: tel-02325512
https://theses.hal.science/tel-02325512

Submitted on 22 Oct 2019

NNT : 2019SACLX047

THÈSE DE DOCTORAT

de

L'UNIVERSITÉ PARIS-SACLAY

École doctorale de mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : École Polytechnique

Établissement d'accueil : Commissariat à l'énergie atomique et aux énergies alternatives (CEA)

Laboratoire d'accueil : Centre de mathématiques appliquées (UMR 7641 CNRS)

Spécialité de doctorat : Mathématiques aux interfaces

Présentée et soutenue par **QINGQING FENG**

Développement d'une méthode d'éléments finis multi-échelles pour les écoulements incompressibles dans un milieu hétérogène

Date et lieu de soutenance : le 20 Septembre 2019 à Saclay

Après avis des rapporteurs :
FRÉDÉRIC LEGOLL (Laboratoire Navier, École des Ponts ParisTech)
ALEXEI LOZINSKI (LmB, Université de Franche-Comté)

Jury de soutenance :

GRÉGOIRE ALLAIRE	Professeur (CMAP, École Polytechnique) Directeur de thèse
ALEXANDRE ERN	Professeur (CERMICS, École des Ponts ParisTech) Président
ALINE LEFEBVRE-LEPOT	Chargée de recherche (CMAP, École Polytechnique) Examinatrice
ALEXEI LOZINSKI	Professeur (LmB, Université de Franche-Comté) Rapporteur
FRÉDÉRIC LEGOLL	Professeur (Laboratoire Navier, École des Ponts ParisTech) Rapporteur
PASCAL OMNES	Directeur de recherche (DEN/DM2S/STMF, CEA) Encadrant

Acknowledgements

Undertaking this PhD has been a truly life-changing experience for me. I am deeply grateful to some people who did their best to put me on the right track.

First of all, I would like to express my sincere gratitude to my doctoral supervisor Dr. Grégoire ALLAIRE, for his invaluable research support and academic advice. I appreciated particularly his extensive personal and professional guidance and provision of precious scientific knowledge. I thank him for his insightful comments and timely guidance in all the time of research and writing of this dissertation. He also demonstrated what a brilliant and hard-working scientist can accomplish.

Besides my doctoral supervisor, I would like to express my great appreciation to Dr. Pascal OMNES, my industrial supervisor at CEA, for his patience, scientific rigor and invaluable expertise. He was assigned as my supervisor at the middle of this research but he tried his best to give me the best guidance possible. His sincere support and encouragement has helped me to get through the hardest time of this research. I have especially learned from him the analytical rigor which I will benefit from for all of my life.

I would like to offer my special thanks to Francis KLOSS, who generously give his time to offer me a great assistance in handling the software *SALOME*. Thanks to Dr. Maria-Adela PUSCAS, with whom I worked during the first phase of this research, especially for her support in handling the software *TrioCFD*. I deeply thank Pierre LEDAC who shared his practical knowledge to help me solve technical problems in *TrioCFD*. I am also very grateful to Dr. Pierre-Emmanuel ANGELI, for providing his valuable expertise in fluid mechanics and useful suggestions in simulation tools. Many thanks to Dr. Erell JAMLOT, for offering me generously her expertise in finite element methods.

I would like to express my great appreciation to Dr. Alexei LOZINSKI and Dr. Frédéric LEGOLL for generously lending me their expertise, giving insightful comments and suggestions which helped me develop a broader perspective to my thesis. I would like to thank them to have accepted to be the referee and be a member of the jury of this dissertation.

I cannot forget dear colleagues Emmanuel KUIDJO KUIDJO and Antoine DU CLUZEAU who did their PhD in the same office for three years. Their encouragement and suggestions helped me to endure the frequent frustration in hard times. It was also a great delight to share each joyful moment and accomplishment with them. I would like to thank all others in the laboratory LMSF and LGLS. The administrative staffs of CEA and of École Polytechnique are memorable not only for their prompt support but also for kind care throughout my dissertation: Violette LEDUC of CEA, Nasséra NAAR and Alexandra NOIRET of CMAP.

I would like to thank my dear friends who encouraged and cheered me on in difficulties and celebrated each accomplishment with me. Finally I would like to express my deep gratitude to my parents for their unconditional trust, timely encouragement and endless support. It was their love which raised me up in difficulties and helped me get through frustration despite the long distance between us.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Challenges and motivation

The sustainable development of the world's energy sector relies on an extensive use of nuclear energy and nuclear related technologies. Nowadays, more than a dozen advanced reactor designs are in various stages of development and nuclear reactor designs of Generation IV and Generation V+ are being actively researched. An accurate prediction of the thermalhydraulic behaviors of the coolant through fuel assemblies is essential for reactor design. Thermalhydraulic behaviors include heterogeneous single-phase and multi-phase flows, the steam/gas mixing and stratification, pressure drops, temperature distributions and so on. Thermalhydraulic studies aim to achieve the maximum safety and efficiency levels in operational states and in a wide range of accident conditions while optimizing the energy generation cost.

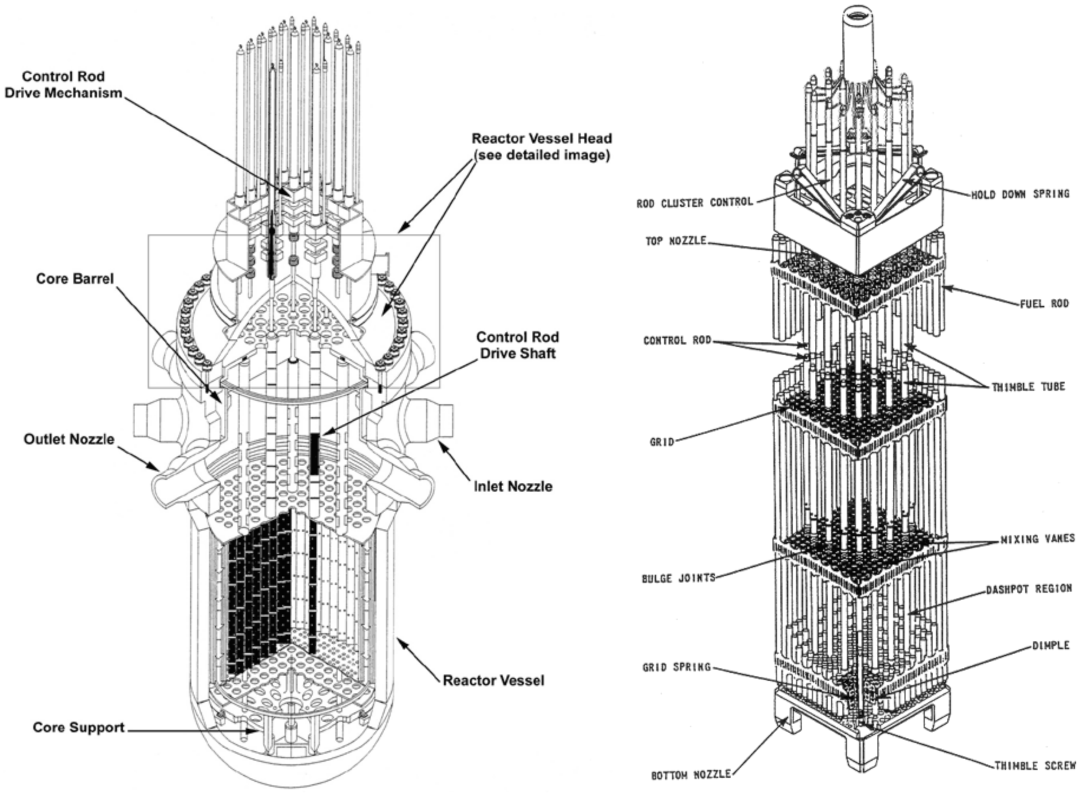


FIGURE 1.1: Schematic representation of a Pressurized Water Reactor (PWR) and a fuel assembly (image from http://www.nrc.gov)

The thermal hydraulic safety analyses of nuclear reactor are performed in two ways. First, in the system level, the System Thermal-Hydraulic codes (STH) such as RELAP5 [127], CATHARE

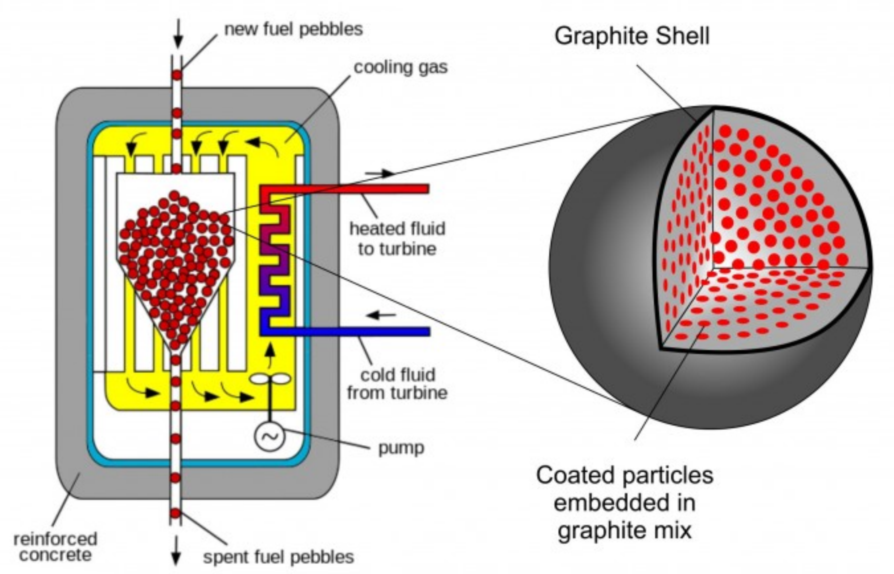


FIGURE 1.2: An illustration of a Pebble Bed Reactor (image from https://en.Wikipedia.org)

[35], ATHLET [27] etc. are developed for the analysis of integral thermalhydraulic features in the nuclear reactor system. They are widely used for safety analysis aiming to assess and improve the safety of existing and new nuclear reactors. They are generally based on averaged and one-dimensional (1D) form of mass, momentum and energy balance equations and some empirical correlations. However, STH software are generally not capable to present a comprehensive and systematic overview of three-dimensional (3D) parameters in the reactor, while single-phase Computational Fluid Dynamics (CFD) software are qualified for a wider range of applications in nuclear reactors, especially complex 3D phenomena. In this context, the coupling of STH and CFD software that cover different scales seems to be promising and a lot of work has been reported, such as the coupling of RELAP5 and FLUENT in [108, 114] and the reference therein. In the Commissariat à l'énergie atomique et aux énergies alerternatives (CEA), active research work has been carried out for the coupling of internal software CATHARE with TrioCFD [139], see [30, 71, 124]. However, the coupling of STH and CFD software is not mature and the performance needs to be further improved.

Second, in the core scale, most of the nuclear reactor cores are designed based on the sub-channel analysis codes, such as COBRA [128] and FLICA [138]. They can estimate the thermal-hyrdaulic safety margins of nuclear reactor core under different steady state and transient operating conditions. The reader can refer to [115] for a rather detailed list of existing sub-channel thermal hydraulic codes. These codes solve conservation equations on some specified control volumes and one dimensional control volumes are connected in both axial and radial directions to get the three dimensional effect of the core. The accuracy of sub-channel codes are greatly improved by taking into account the flow mixing between fuel assembly sub-channel. However, a correct formulation of conservation equations and a good knowledge of the mixing process is necessary in order to obtain reliable predictions of thermohydraulic behaviors in nuclear reactor core.

Theoretically, the CFD analysis can provide detailed three-dimensional thermalhydraulic behaviors of the reactor core. However, in practice, as shown in the left of Figure 1.1, the scales of interest spread over several orders of magnitude in the Reactor Pressure Vessel (RPV) of a Pressurized Water Reactor (PWR). The overall length of the RPV (including closure head and nozzles) is about $14\,\mathrm{m}$. The reactor core is about $4\,\mathrm{m}$ to $5\,\mathrm{m}$ in diameter and consists of between

157 and 200 fuel assemblies shown in the right of Figure 1.1. The fuel assembly is about $4\,\mathrm{m}$ to $5\,\mathrm{m}$ long and consists of fuel rods generally bundled in a square array of 14×14 to 17×17. The thickness of the mixing vanes in the spacer mesh is about $1\,\mathrm{mm}$ and the diameter of each fuel rod is about $1\,\mathrm{cm}$. Therefore the RPV contains various scales and the global scale is about 10^4 times larger than the local scale.

In the 4-Loop Westinghouse PWR multi-physics reduced model presented in [111], only one fourth of the 193 fuel assemblies are taken into account and there are in total 13,944 fuel rods, 434 spacers and 148,224 mixing vanes. A very complexe mesh consisting of about 1.2 billion elements is used to resolve all the geometric details in the RPV. Thermohydraulic simulations and multi-physics simulations in the model show that the full core performance is effectively affected by localized phenomena. This is thus a typical multiscale problem. It is predictable that an extremely fine mesh is necessary to resolve all the scales in the RPV. However, direct simulations on such a mesh require a massive amount of computer memory and computing time, which can easily exceed today's computing power. Parallel computing can relieve to some degree the situation but the size of the problem is not reduced actually.

The Pebble Bed Reactor (PBR) [126] is also a multiscale system. A schematic representation of the PBR and a fuel element are shown in Figure 1.2. The reactor core contains about 600,000 spherical fuel and moderator elements and each fuel element is about $6\,\mathrm{cm}$ in diameter. An extremely fine mesh is thus required in a direct CFD simulation to resolve the large number of small fuel elements in the reactor core. Consequently, a direct CFD simulation of the entire reactor core is extremely expensive or even impossible with modern computing power.

Given the complexity of the reactor core geometry (see Figure 1.1 and Figure 1.2) and limited computing power, most CFD studies employ homogenized core models to reduce the computational cost and to minimize the computing time. One typical method is the porous media approach where the reactor core is simplified to a single porous medium. Some CFD software such as ANSYS CFX [36] provide porous media model for flows in porous region. Parameters for the porous media model are estimated from CFD analysis of a single fuel assembly or from empirical studies of the reactor core. The reader can refer to [38] for a concrete example of this approach. At CEA, this approach is explored in several thesis, such as [20, 37]. These works employ the volume averaging approach combined with the Representative Elementary Volume (REV) concept which is presented in section 1.4. The macroscopic simulation could provide preliminary results but the performance of the method depends on parameters used in the porous media, which are sometimes empirical.

The difficulty of performing a direct CFD simulation of the entire reactor core lies in the fact that it is too expensive to resolve all the scales. But from an application perspective, it is often sufficient to predict important macroscopic behaviors. As a consequence, it is desirable to develop some methods which are able to capture the effects of microscopic scales on macroscopic scales, but do not resolve all the microscopic features. Along this direction, many methods such as upscaling methods, multiscale methods and so on have been proposed for multiscale problems over the years. These methods attempt to capture fine-scale phenomena on coarse meshes and microscopic scales below the coarse mesh scale are resolved by incorporating local computations into a global formulation defined only on the coarse mesh. These methods are thus able to provide solutions with the desired accuracy at reasonable computational costs. An overview of these approaches is given in section 1.4.

1.2 Flow problems in heterogeneous media

Modeling of flows in heterogeneous perforated domains is a topic of significant interest in a wide range of engineering and technical applications, such as reservoir engineering, flows through

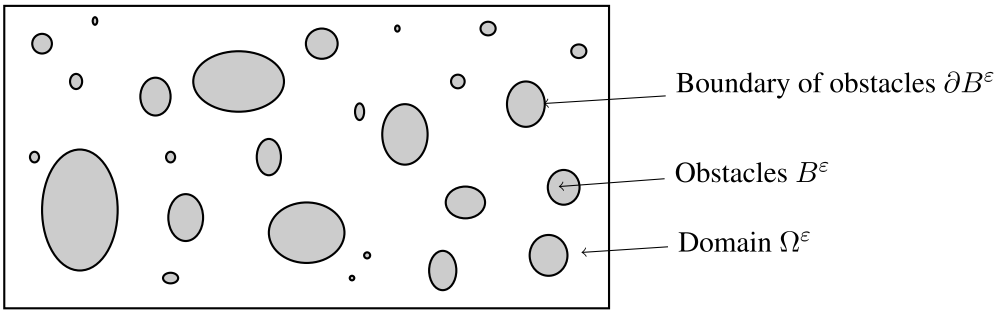


FIGURE 1.3: Schematic description of the perforated domain $\Omega^\varepsilon = \Omega \setminus \bar{B}^\varepsilon$

vuggy or fractured porous media, and so on. As shown in Figure 1.3, a perforated domain Ω^ε is a flow domain with voids left by solid obstacles B^ε.

Let $\Omega \subset \mathbb{R}^d$ be a regular bounded open set, with the dimension $d \in \{2, 3\}$. As shown in Figure 1.3, we divide the domain Ω into a fixed solid part B^ε and its complementary fluid part Ω^ε, where $\varepsilon > 0$ denotes the length-scale of the heterogeneities. The solid part is impermeable. In general, B^ε can be a periodic array or obstacles or a set of randomly placed obstacles, each with a diameter of ε and separated by a distance at the order of ε. For example, B^ε can be fuel rods arranged periodically in a nuclear fuel assembly or rock matrices in sub-surface flows.

This thesis is devoted to solving incompressible single-phase flow problems in very heterogeneous media such as the RPV. Now let us first review the definition of different flow problems.

Navier-Stokes problem On the perforated domain Ω^ε, the incompressible Navier-Stokes equations [134] can be used to model fluid flows in Ω^ε. The steady-state incompressible Navier-Stokes problem with homogeneous Dirichlet boundary condition is to find the velocity $\boldsymbol{u} : \Omega^\varepsilon \to \mathbb{R}^d$ and the pressure $p : \Omega^\varepsilon \to \mathbb{R}$ solutions to:

$$\begin{cases} -\mu\Delta \boldsymbol{u} + \rho\left(\boldsymbol{u} \cdot \nabla\right)\boldsymbol{u} + \nabla p = \boldsymbol{f} \text{ in } \Omega^\varepsilon \\ \text{div } \boldsymbol{u} = 0 \text{ in } \Omega^\varepsilon \\ \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap \partial\Omega^\varepsilon \\ \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial\Omega \cap \partial\Omega^\varepsilon \end{cases} \tag{1.1}$$

where $\boldsymbol{f}$ is a given force, μ is the dynamic viscosity and ρ is the flow density.

The nonlinear convective term $\rho\left(\boldsymbol{u} \cdot \nabla\right)\boldsymbol{u}$ in the Navier-Stokes equations increases the difficulties in mathematical analysis and numerical solutions of these equations. The Reynolds number defined below is a dimensionless quantity which is defined as the ratio of inertial forces to viscous forces within a fluid. It quantifies the relative importance of these two forces and predict flow patterns.

$$Re = \rho U_c L/\mu$$

where U_c is a scalar characteristic velocity and L is a characteristic length.

Stokes problem The Reynolds number of flows in porous media is usually small and the inertial forces can be neglected compared to the viscous forces. This simplification leads to the Stokes problem which is a suitable model of the physical process at pore scales. Stokes flows [134] exist in diverse engineering practices such as reservoir engineering. The steady-state Stokes problem with homogeneous Dirichlet boundary condition is to find the velocity $\boldsymbol{u} : \Omega^\varepsilon \to \mathbb{R}^d$

and the pressure $p : \Omega^\varepsilon \to \mathbb{R}$ solutions to:

$$\begin{cases} -\mu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \text{ in } \Omega^\varepsilon \\ \text{div } \boldsymbol{u} = 0 \text{ in } \Omega^\varepsilon \\ \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap \partial \Omega^\varepsilon \\ \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial \Omega \cap \partial \Omega^\varepsilon \end{cases} \tag{1.2}$$

where $\boldsymbol{f}$ is a given force, μ is the dynamic viscosity and complemented with appropriate boundary conditions.

Oseen problem For steady incompressible flows, the Oseen's approximation [69, 70, 132] is a linearization of Navier-Stokes equations. The nonlinear inertial term $\rho \left(\boldsymbol{u} \cdot \nabla \right) \boldsymbol{u}$ is substituted by the linear Oseen term $\rho \left(\boldsymbol{U}_o \cdot \nabla \right) \boldsymbol{u}$, with $\boldsymbol{U}_o$ a known velocity. For flows around a sphere, the Stokes solution is not valid at large distances from the solid body, because the advective terms are not negligible compared to the viscous terms. Oseen provided a cure to Stokes solution by partly accounting for the inertia terms at large distances. Near the body both Stokes and Oseen approximations have the same order of accuracy. However, the Oseen approximation is better in the far field where the velocity is only slightly different than $\boldsymbol{U}_o$.

The steady Oseen problem with homogeneous Dirichlet boundary condition is to find the velocity $\boldsymbol{u} : \Omega^\varepsilon \to \mathbb{R}^d$ and the pressure $p : \Omega^\varepsilon \to \mathbb{R}$ solutions to:

$$\begin{cases} -\mu \Delta \boldsymbol{u} + \rho \left(\boldsymbol{U}_o \cdot \nabla \right) \boldsymbol{u} + \nabla p = \boldsymbol{f} \text{ in } \Omega^\varepsilon \\ \text{div } \boldsymbol{u} = 0 \text{ in } \Omega^\varepsilon \\ \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap \partial \Omega^\varepsilon \\ \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial \Omega \cap \partial \Omega^\varepsilon \end{cases} \tag{1.3}$$

where $\boldsymbol{U}_o$ a given Oseen velocity, $\boldsymbol{f}$ is a given function, μ is the dynamic viscosity, ρ is the flow density and complemented with appropriate boundary conditions.

The Oseen flow plays an important role in characterizing the asymptotic structure of steady solutions to the Navier–Stokes problems at large distances from solids. In the pre-CFD era, Oseen equations generated significant interest since its solutions offered valuable quantitative insights into properties of flows past solid bodies at low Reynolds numbers. Then the CFD makes it possible to solve the Navier-Stokes equations numerically and the interest in the Oseen approximation subsided. The Oseen system is still occasionally used as a testbed for validating numerical approaches, such as stabilization of finite element discretizations [8], or artificial boundary conditions on truncated computational domains [146] or weak Galerkin finite element method [110] and so on. Besides, after applying a time discretization and a linearization, many solution algorithms for Navier-Stokes equations are reduced to solving a sequence of Oseen equations if the advection is treated explicitly. This is why Oseen equations are often used as a first step towards the analysis of the full Navier-Stokes equations.

In an heterogeneous medium with the characteristic length $\varepsilon > 0$, the number of degrees of freedom and the computational cost to solve above flows problems with classic numerical methods, such as the finite element method, is in the order of $\mathcal{O}\left(\text{vol}(\Omega)\varepsilon^{-d}\right)$ with $\text{vol}(\Omega)$ the volume of Ω. It becomes easily prohibitively expensive for small values of ε.

Darcy's law The widely used model for flows in porous media is the Darcy equation [86]. It is an elliptic partial differential equation which describes the effective pressure and the effective velocity of a saturated fluid in porous media. Many publications such as [10, 12, 14, 86] have derived the Darcy's law for incompressible viscous fluid flows in porous media. Starting from the steady Stokes equations in a periodic porous medium, with a no-slip (Dirichlet) boundary condition on the solid pores (or obstacles), [10] shows that Darcy's law can be rigorously obtained by periodic homogenization using the two-scale convergence method. The derivation process is presented in detail in section 1.3.

For any given force term $\boldsymbol{f}$ and the effective permeability tensor $a^0 : \Omega \to \mathbb{R}^{d\times d}$, the Darcy's law is to find the averaged quantity $\left(\boldsymbol{u}^0, p^0\right)$ which is solution of

$$\begin{cases} \boldsymbol{u}^0 = \frac{\kappa^0}{\mu}\left(\boldsymbol{f} - \nabla p^0\right) \\ \operatorname{div} \boldsymbol{u}^0 = 0 \\ \boldsymbol{u}^0 \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega \end{cases} \tag{1.4}$$

where the the effective permeability tensor κ^0 depends only on the microstructure of the porous media (neither on external forces $\boldsymbol{f}$ nor on the physical properties μ and ρ of the fluid). The Darcy equation can be discretized and solved with classic numerical methods at a low cost no matter the value of ε. Of course, the homogenization of flow models more complicated than the Stokes equations can lead to various Darcy's law. However, Darcy's law is only valid for creeping flows and we obtain only averaged solutions, i.e., oscillating or fluctuating features are not visible in the solution.

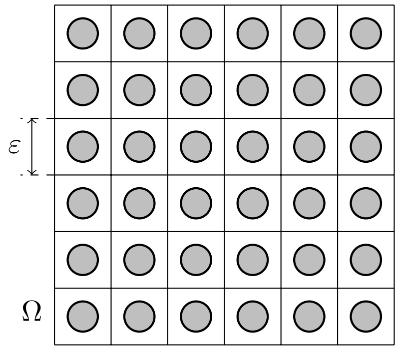


FIGURE 1.4: An illustration of a periodic domain with periodicity ε

1.3 The homogenization theory

We now briefly introduce the homogenization theory, which is the basic building block of most multiscale methods. For a deeper understanding of the homogenization theory, the reader can refer to [16, 29, 86, 98]. Given the microscopic description of a problem, homogenization is the process of seeking a macroscopic or effective description via asymptotic analysis. Homogenization was first developed for periodic media but can be applied to any kind of heterogeneous media. Besides, new models in various multiscale methods are justified as the homogenized limits of complex microscopic equations. Based on [10], this section presents the homogenization via a simple diffusion problem in a periodic medium.

1.3.1 Problem setting

As shown in Figure 1.4, let Ω be a periodic domain which is a bounded open set in $\mathbb{R}^d$ with periodicity ε (ε is assumed to be very small compared to the size of the domain). The scaled unit periodic cell is $Y = (0,1)^d$. The conductivity tensor in Ω is $A\left(\frac{x}{\varepsilon}\right)$ where $A(y)$ is Y-periodic and satisfy the coerciveness assumption:

$$\alpha|\xi|^2 \leq \sum_{i,j}^{d} A_ij\left(y\right)\xi_i\xi_j \leq \beta|\xi|^2 \quad \forall\xi \in \mathbb{R}^d,\ \forall y \in Y$$

with $\beta \geq \alpha > 0$.

Denoting f the source term (a scalar function defined in Ω) and imposing a Dirichlet boundary condition, the model problem of conductivity reads

$$-\text{div}\left(A\left(\frac{x}{\varepsilon}\right)\nabla u_\varepsilon\right) = f \text{ in } \Omega \tag{1.5}$$

$$u_\varepsilon = 0 \text{ on } \partial\Omega \tag{1.6}$$

where u_ε is the unknown modeling the electrical potential or the temperature.

Solving the above problem with conventional numerical methods such as finite element methods requires a mesh with element size $h \ll \varepsilon$. If ε is very small, the mesh is extremely fine and the discrete problems have a large number of degrees of freedom. Therefore it is too expensive or impossible to solve such problems due to limited computing power and storage space. It is thus desirable to homogenize the properties of Ω and compute an approximation of u_ε on a coarse mesh.

We mention that there is a difference between the traditional physical approach of homogenization and the mathematical theory of homogenization. The volume averaging approach combined with the representative volume element (RVE) method [100] consists in taking a sample of the heterogeneous medium of size much larger than the heterogeneities but still much smaller than the medium. This sample domain is called RVE. The ∇u_ε and the flux $A\left(\frac{x}{\varepsilon}\right)\nabla u_\varepsilon$ are averaged over the RVE and we denote their average as ξ and σ respectively. The effective conductivity A^* of the RVE is defined by the linear relationship $\sigma = A^*\xi$. It turns out that the averaged stored energy $A\left(\frac{x}{\varepsilon}\right)\nabla u_\varepsilon \cdot \nabla u_\varepsilon$ equals to the effective energy $A^*\xi \cdot \xi$. This definition is very intuitive and it is not clear whether the effective tensor A^* is defined correctly. In particular, it may depend on the choice of source term f, the sampling size or boundary conditions. For more concrete examples of the applications of the RVE method, see [20, 37].

As pointed out in [10], the mathematical theory of homogenization works completely differently. Rather than considering a single heterogeneous medium with a fixed length scale, the problem is first embedded in a sequence of similar problems for which the length scale ε, becoming increasingly small, goes to zero. Then, an asymptotic analysis is performed as ε tends to zero and the conductivity tensor of the limit problem is called the effective or homogenized conductivity. This approach has the advantage of defining uniquely the homogenized properties. Further, the approximation made by using effective properties instead of the true microscopic coefficients can be rigorously justified by quantifying the resulting error.

In the case of a periodic medium Ω, the asymptotic analysis of the equation (1.5) is quite simple and the solution u_ε is written as a power series in ε

$$u_\varepsilon = \sum_{i=0}^{\infty} \varepsilon^i u_i$$

The first term u_0 of this series will be identified as the solution of the homogenized equation whose effective conductivity A^* can be computed exactly. It turns out that A^* is a constant tensor describing a homogeneous medium, which is independent of f and the boundary conditions. Therefore, numerical computations on the homogenized equation do not require a fine mesh since the heterogeneities of size ε have been averaged out. This homogenized tensor A^* is almost never a usual average (arithmetic or harmonic) of $A(y)$. Various estimates will confirm this asymptotic analysis by telling in which sense u_ε is close to u_0 as ε tends to zero.

1.3.2 Two-scale asymptotic expansions

We consider the following two-scale asymptotic expansion for the solution u_ε of (1.5).

$$u_\varepsilon(x) = \sum_{i=0}^{\infty} \varepsilon^i u_i\left(x, \frac{x}{\varepsilon}\right)$$

where each term $u_i(x, y)$ is a function of both x and y, periodic in y with period $Y = (0,1)^d$ (u_i is a Y-periodic function with respect to y). This series is plugged into the equation and the following derivation rule is used:

$$\nabla\left(u_i\left(x, \frac{x}{\varepsilon}\right)\right) = \left(\varepsilon^{-1}\nabla_y u_i + \nabla_x u_i\right)\left(x, \frac{x}{\varepsilon}\right)$$

where ∇_x and ∇_y denote the partial derivative with respect to the macroscopic variable x and the microscopic variable y. For example,

$$\nabla u_\varepsilon(x) = \varepsilon^{-1}\nabla_y u_0\left(x, \frac{x}{\varepsilon}\right) + \sum_{i=0}^{\infty} \varepsilon^i \left(\nabla_y u_{i+1} + \nabla_x u_i\right)\left(x, \frac{x}{\varepsilon}\right)$$

Therefore (1.5) can be written as a series of ε

$$\begin{aligned}
&- \varepsilon^{-2}\left[\mathrm{div}_y A\nabla_y u_0\right]\left(x, \frac{x}{\varepsilon}\right)\\
&- \varepsilon^{-1}\left[\mathrm{div}_y A\left(\nabla_x u_0 + \nabla_y u_1\right) + \mathrm{div}_x A\nabla_y u_0\right]\left(x, \frac{x}{\varepsilon}\right)\\
&- \varepsilon^{0}\left[\mathrm{div}_x A\left(\nabla_x u_0 + \nabla_y u_1\right) + \mathrm{div}_y A\left(\nabla_x u_1 + \nabla_y u_2\right)\right]\left(x, \frac{x}{\varepsilon}\right)\\
&- \sum_{i=0}^{\infty} \varepsilon^{i}\left[\mathrm{div}_x A\left(\nabla_x u_i + \nabla_y u_{i+1}\right) + \mathrm{div}_y A\left(\nabla_x u_{i+1} + \nabla_y u_{i+2}\right)\right]\left(x, \frac{x}{\varepsilon}\right)\\
&= f(x)
\end{aligned}$$

Identifying each coefficient of the above equation as an individual equation yields a cascade of equations. It turns out that the first three equations are enough for our purpose. The ε^{-2} equation is

$$-\mathrm{div}_y A(y)\nabla_y u_0(x, y) = 0$$

which is nothing else than an equation in the unit cell Y with periodic boundary condition. In this equation, y is the variable, and x plays the role of a parameter. There exists a unique solution of this equation up to a constant. This implies that u_0 is a function that does not depend on y, i.e. there exists a function $u(x)$ such that

$$u_0(x, y) \equiv u(x)$$

Since $\nabla_y u_0 = 0$, the ε^{-1} equation is

$$-\mathrm{div}_y A(y)\nabla_y u_1(x, y) = \mathrm{div}_y A(y)\nabla_x u(x) \tag{1.7}$$

which is an equation for the unknown u_1 in the periodic unit cell Y. It is a well-posed problem, which admits a unique solution up to a constant, as soon as the right hand side is known.

This equation allows one to compute u_1 in terms of u and $u_1(x,y)$ depends linearly on the first derivative $\nabla_x u(x)$. Finally the ε^0 equation is

$$-\text{div}_y A(y)\nabla_y u_2\,(x,y) = \text{div}_y A(y)\nabla_x u_1 + \text{div}_x A(y)\left(\nabla_y u_1 + \nabla_x u\right) + f(x)$$

which is an equation for the unknown u_2 in the periodic unit cell Y. The above equation admits a solution if a compatibility condition (the *Fredholm alternative* in Lemma 1.3.1) is satisfied. Indeed, integrating the left hand side of the above equation over Y and using the periodic boundary condition for u_2, we obtain

$$\int_Y \text{div}_y A(y) u_2\,(x,y)\ dy = \int_{\partial Y} \left[A(y)\nabla_y u_2\,(x,y)\right]\cdot \boldsymbol{n} = 0$$

This implies that

$$\int_Y \left[\text{div}_y A(y)\nabla_x u_1 + \text{div}_x A(y)\left(\nabla_y u_1 + \nabla_x u\right) + f(x)\right]\ dy = 0$$

which simplifies to

$$-\text{div}_x \left(\int_Y A(y)\left(\nabla_y u_1 + \nabla_x u\right)\ dy\right) = f(x) \text{ in } \Omega \tag{1.8}$$

Since $u_1(x,y)$ depends linearly on $\nabla_x u(x)$, the above equation is simply an equation for $u(x)$ involving only the second order derivatives of u.

The cell and homogenized problems In order to compute u_1 and to simplify (1.8), we introduce the cell problems. We denote $(e_i)_{1\leq i\leq d}$ the canonical basis of $\mathbb{R}^d$. For each vector e_i, we consider the following conductivity problem in the periodic unit cell:

$$\begin{cases} -\text{div}_y A(y)\left(e_i + \nabla_y \omega_i\,(y)\right) = 0 & \text{in } Y \\ y \to \omega_i\,(y) & Y\text{-periodic} \end{cases}$$

where $\omega_i\,(y)$ is the local variation of potential or temperature created by an averaged gradient e_i. The existence of a solution ω_i is guaranteed by the following theory.

Lemma 1.3.1 (The Fredholm alternative)**.** *Let $f\,(y) \in L^2\,(Y)$ be a periodic function. There exists a solution in $H^1(Y)$ (unique up to an additive constant) of*

$$\begin{cases} -div\ A(y)\nabla\omega(y) = f & in\ Y \\ y \to \omega(y) & Y - periodic \end{cases}$$

if and only if $\int_Y f(y)dy = 0$.

By linearity, we can compute $u_1\,(x,y)$ solution of (1.7) in terms of (x) and $w_i(y)$

$$u_1\,(x,y) = \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}\,(x)\,\omega_i\,(y)$$

Inserting this expression into (1.8), we obtain the homogenized equation for u that we supplement with a Dirichlet condition on $\partial\Omega$,

$$\begin{cases} -\text{div}_x A^* \nabla_x u(x) = f(x) & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases}$$

The homogenized conductivity A^* is defined by its entries

$$A^*_{ij} = \int_Y \left[(A(y)\nabla_y \omega_i) \cdot e_j + A_{ij}(y) \right] \, dy = \int_Y A(y) \left(e_i + \nabla_y \omega_i \right) \cdot \left(e_j + \nabla \omega_j \right) \, dy$$

The constant tensor A^* describes the effective or homogenized properties of the heterogeneous material $A\left(\frac{x}{\varepsilon}\right)$. A^* does not depend on the choice of the domain Ω, source term f or boundary condition on $\partial\Omega$. Therefore the solution u_ε can be approximated as detailed in [10] by:

$$u_\varepsilon(x) \approx u(x) + \varepsilon \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}(x)\, \omega_i\left(\frac{x}{\varepsilon}\right)$$

where the term $\varepsilon \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}(x)\, \omega_i\left(\frac{x}{\varepsilon}\right)$ is the corrector term. This term improves the approximation of u_ε. When ε is small, the corrector term can be neglected if one is interested in the value of u_ε. Meanwhile, if one is interested in the value of ∇u_ε, the corrector term should be taken into account since it is of the same order as the homogenized gradient ∇u,.

1.4 Literature overview

A broad range of important scientific and engineering problems are multiscale problems, which arise due to high contrasts of the system or due to spatial heterogeneity of the domain. Typical examples include modeling of composite materials, flows in porous media and turbulent transport in high Reynolds number flows, and so on. In composite materials, the heterogeneity is represented by multiscale fluctuations in the thermal conductivity or elastic property. In turbulent transport problems, the convective velocity field fluctuates randomly and contains many scales depending on the Reynolds number of the flow.

In the last section, we have presented homogenization in periodic porous media in order to solve multiscale problems at reduced computational costs. With homogenization, we obtain only the averaged solution by solving the homogenized equations. However, we expect not only to obtain the homogenized behavior of the real solution, but also take into account the fluctuations related to multiple scales of the problem. Along this direction, many model reduction techniques have been proposed over the years based on the homogenization theory. These methods are also called numerical homogenization methods. In this section, we give an overview of such techniques in the literature and the state-of-the-art of their applications to solve flow problems in heterogeneous media.

1.4.1 Numerical homogenization methods

Homogenization of fluid flow in porous media is the mathematical description of the asymptotic behavior of the flow when $\varepsilon \to 0$ (pore scale or the characteristic length of heterogeneity). This mathematical theory was first described in [130] for the Stokes equations in periodic porous domain. It was shown that the Stokes equations can be derived into an effective Darcy equation. The effective permeability can be computed from a microscopic computation using the Stokes equations in a sampling domain by taking into account the porous structure. These findings were rigorously established by [133] and then generalized by[11, 13–15] and [66] to other flow problems in porous media. In [18, 19], homogenization of conductive and radiative heat transfer problems in periodic porous media are performed recently. However, its applications are limited by the restrictive assumptions on the media or the coefficient: **scale separation** and **periodicity**. Besides, it is expensive to use for problems with many separate scales since the computational cost grows exponentially with the number of scales [87]. However, homogenization theory has triggered the development of numerical methods that solve the homogenized equations numerically.

The Upscaling Method is based on the homogenization theory. It is the process of representing the system on a coarser scale by defining the average or effective macroscopic parameters. The method has been applied to solve flow problems in highly heterogeneous media or composite materials in [39, 49, 68, 93, 94, 144]. However, the application of upscaling methods is still constrained by the assumptions of homogenization. Besides, it is difficult to have a priori estimates of the errors when complex flow processes in general heterogeneous media are investigated using coarse models constructed via simplified settings.

For situations where the homogenization theory does not work, its idea can still be exported to other numerical approaches. Instead of computing the effective properties as in upscaling approaches, one computes multiscale basis functions in multiscale methods. The problem is fully resolved at the fine scale (smaller than ε) to capture significant local features and a global coarse problem is solved at the coarse scale (much larger than ε) by incorporating fine scale computations. Computational efficiency comes from a divide-and-conquer approach: small, localized fine scale problems are easily solved and the global coarse problem has only a few degrees of freedom per coarse element. We give below a list of main multiscale methods and some references. The list is certainly not exhaustive since there is a large number of works in this area.

The Variational Multiscale Method began with the work of [91, 92]. The idea of the method is decompose the solution into two groups: the resolved (coarse) and the unresolved (fine) parts. The fine scale part can further be decomposed into independent local unknowns. Since local unknowns are independent, the global problem is decomposed into small local problems which can be solved independently. The solutions of local problems can be coupled by a coarse scale problem defined by elements in the coarse scale part. The solution of coarse scale problem is assumed to be linear which affects the accuracy of the method. This approach was defined for the mixed case in [22–25].

The Heterogeneous Multiscale Method (HMM) was first proposed in [57] based on homogenization and reviewed in [7, 58, 83]. The name “heterogeneous” was used to emphasize the “multi-physics” applications that it targets. HMM is in fact a framework for linking models at different scales. The homogenized problem is solved by conventional methods such as finite element methods on a coarse mesh. When assembling the discrete problems, local problems are solved and solutions are used in the numerical integration. In this way, microscopic features are taken into account in the homogenized problem. HMM are particularly attractive when the information of the media is only available in some local representative volumes [106]. The work of [4–6] proposed recently a variant of HMM for Stokes flows in porous media. This approach is based on the Darcy-Stokes coupling described by the homogenization theory.

One of the most popular multiscale approaches is the Multiscale Finite Element Method (MsFEM), which is the primary concern of this thesis. We review various MsFEMs in detail and discuss their advantages and limitations in subsection 1.4.2. Another well-known multiscale approach is the Multiscale Finite Volumes Method (MsFVM) [31, 51, 72, 96, 97]. This approach aims mainly at subsurface flow simulations. These approaches share some of the main ideas of MsFEM. But multiscale basis functions are constructed on a dual coarse mesh. These basis functions guarantee the mass conservative at both the coarse and the fine scales. This makes the method particularly attractive for transport problems.

There are many other multiscale methods, such as Mortar Multiscale Finite Element Methods [74], Multigrid Methods [26, 112, 116, 140, 145], Multiscale Discontinuous Galerkin Methods [1, 3], Iterative-Multi-Scale-Finite-Volume Method [31] and so on. We mention that [31] presented the application of Iterative-Multi-Scale-Finite-Volume Method to solve the pressure Poisson equation arising in numerical methods for the simulation of incompressible flows with the immersed-interface method.

At last, we mention that domain decomposition methods as described in [56, 75, 137] are quite different from multiscale methods such as MsFEM. In domain decomposition methods, local problems are solved iteratively to obtain an accurate approximation of the fine-scale solution. These iterations guarantee the convergence of the method. The cost of iterations can be high, in particular for multiscale problems. On the contrary, multiscale methods aim to capture fine-scale properties as accurately as possible and avoid solving local problems many times.

Development of multiscale numerical methods at CEA In fact, multiscale methods were developed for the first time at CEA by [2], for diffusion problems in porous media, in the context of storage of nuclear waste in cement. Several multiscale methods were developed and compared, such as the MsFEM, MsFVM and Multiscale Discontinuous Galerkin Methods. Oversampling techniques are applied in the computation of multiscale basis functions. A multiscale simulation chain was finally built based on the *SALOME* platform [129] and *TRUST* [139], which we present more in Chapter 5 of the thesis. The work of [2] has effectively been, to some degree, a good reference to this thesis since they share some similarities in simulation tools and in multiscale methods.

1.4.2 The Multiscale Finite Element Methods (MsFEMs)

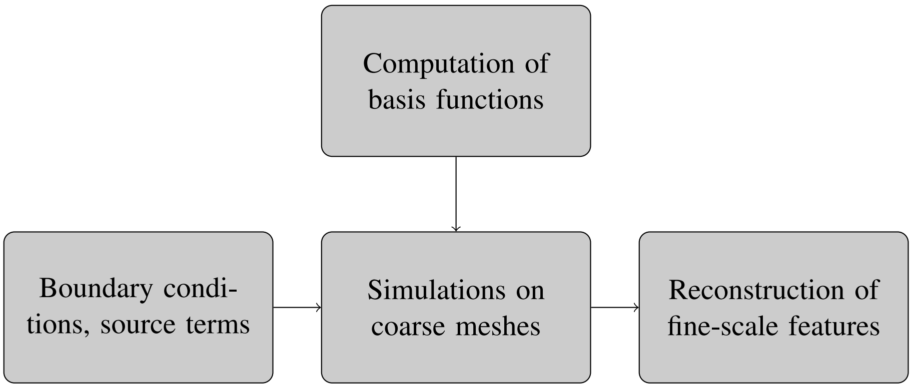


FIGURE 1.5: An illustration of multiscale simulations and advantages

The Multiscale Finite Element Method was first introduced by [87, 88] for numerical solution of multiscale problems that are described by partial differential equations with highly oscillatory coefficients. The main idea of MsFEM is to incorporate the microscopic features of a multiscale problem into some special finite element basis functions (different from the conventional finite element basis functions). Then the effects of microscales on macroscales are captured through the coupling of these special finite element basis functions and a global numerical formulation such as the Galerkin method.

Since then, MsFEMs have experienced major advancements in [2, 17, 40, 60, 61, 63] and the references therein. They have mainly applied MsFEMs for elliptic problems with oscillating coefficients in composite materials and porous media. [17, 85] generalized the original approach which uses linear finite elements (see [87, 88]) to approaches which use higher-order finite elements. Higher-order finite elements can improve the accuracy and the efficiency of the method. [120, 121] provide a priori error estimate of a MsFEM for convection-diffusion problems where both the velocity and diffusion coefficient exhibits multiscale features. MsFEMs have also been generalized to solve nonlinear time-dependent problems [62] in porous media. Many new variants of MsFEMs appeared over the years, such as Mixed Multiscale Finite Element Methods [24, 39, 106] and Generalized Multiscale Finite Element Method (GMsFEM) [44, 46–48, 59, 107] and so on. The work of [45, 47, 48] present a GMsFEM to solve Stokes problems in perforated domains at both the fine and coarse scales. The extension to other flow problems such as Navier-Stokes equations is not studied yet.

MsFEMs can handle cases with or without scale separation and there are no restrictive assumptions as in homogenization. As described in [61], MsFEMs consist of two main elements: multiscale basis functions and a global numerical formulation that couples the multiscale basis functions. The multiscale basis functions are computed on coarse elements and the union of coarse elements form a coarse mesh. Important fine-scale features of the solution are captured by localized multiscale basis functions. Then a global formulation defined on the coarse-mesh couples the multiscale basis functions to provide an accurate solution of the problem. This process guarantees that the effect of fine-scales on coarse-scales are corrected captured. It is well-known that conventional finite element methods use piecewise polynomials as basis functions of the approximation space. These piecewise polynomials need to satisfy some continuity requirements across inter-element interfaces in order to ensure the convergence of the method. For example, the Crouzeix–Raviart element [52] uses piecewise linear polynomials which are only continuous at the midpoint of the interface between adjacent elements.

Therefore MsFEMs can be considered as a finite element method that do not use piecewise polynomial basis functions but some special pre-computed basis functions. MsFEMs can thus be easily implemented within an existing finite element software. As introduced in [87], the main idea is to construct basis functions by solving local problems with prescribed (artificial) boundary conditions. Basis functions can thus incorporate the fine-scale features of the problem. Then multiscale basis functions are then coupled via a global formulation (the coarse-scale problem). This coupling typically requires some overlapping or some parameters for conforming approaches, which may increase the complexity of the method. However, we expect to couple basis functions with as minimal information as possible and keep only a few degrees of freedom in each coarse element. It is thus desired to develop nonconforming multiscale methods.

1.4.2.1 Nonconforming MsFEMs

One great challenge of the MsFEMs is the error due to scale resonance (see [144]), which is characterized by the ratio between the small physical scales of the media and the artificial size of the coarse mesh. Numerical errors become large when the small physical scale is close to the coarse size. [63, 87, 88] revealed two main sources of scale resonance.

One source of scale resonance is the mismatch between the artificial local boundary conditions imposed on the basis functions and the global nature of the oscillatory solution of the differential operator. This origin induces boundary layers in the first order corrector of basis functions. It was pointed out that this error is most significant and it can be reduced by using an oversampling method. The idea of oversampling methods is to solve local problems in a domain larger than the coarse element itself, aiming to reduce resonance errors due to artificial boundary conditions. The multiscale basis functions are obtained by restricting solutions of local problems to the coarse elements. MsFEMs with oversampling techniques are thus nonconforming methods and errors related to the nonconformity was studied in [63]. Various oversampling methods can be found in [43, 61, 64, 72, 84] and the references therein.

The other origin of scale resonance is the mismatch between the size of the coarse mesh and the "perfect" sample size. The "perfect" sample size would be an integer multiple of the period for periodic media. This error can not be entirely removed by oversampling methods, but it is of a lower order. Then [89] proposed a Petrov-Galerkin formulation [99] to remove this error, which leads to a Petrov-Galerkin nonconforming MsFEM.

Another type of nonconforming MsFEM was proposed in [41] for solving a second-order multiscale elliptic problem. In this approach, there are two sources of nonconformity. The first one comes from the fact that nonconforming finite elements are used on the coarse mesh. While constructing multiscale basis functions, Dirichlet linear boundary conditions are imposed on the boundary of coarse elements and in particular, basis functions are imposed to be continuous only at the midpoint of the interface between two neighboring coarse elements. Besides, basis functions are imposed to be zero at the midpoint of edges lying on the boundary of Ω. This finite element element is in fact the conventional Crouzeix–Raviart element [52]. However, oversampling techniques are needed to reduce resonance errors due to the mismatch between the local basis functions and the solution of the global formulation on the boundary of each coarse element. The oversampling technique induces the second source of nonconformity.

Another type of nonconforming MsFEM based on the Crouzeix–Raviart element was proposed in [103] and denoted as Crouzeix–Raviart MsFEM. In this approach, basis functions are constructed related to the interface between neighboring coarse elements. The non-conforming nature of Crouzeix-Raviart element is shown to provide a good flexibility especially when arbitrary patterns of obstacles are considered. Based on this feature, the Crouzeix–Raviart MsFEM imposes only a weak conformity along the interface of adjacent coarse elements, i.e. only the average of the "jump" of basis functions vanish at the interface between neighboring coarse element. This leads to a natural boundary condition on the interface of coarse elements, which relaxes the sensitivity of the method to complex patterns of obstacles, without using oversampling methods. The Crouzeix–Raviart MsFEM is different to to the nonconforming MsFEM [41] presented in the last paragraph. Firstly, the Crouzeix–Raviart MsFEM does not impose linear Dirichlet boundary conditions on the boundary of coarse elements. Secondly, the Crouzeix–Raviart MsFEM does not need to use any oversampling techniques while computing basis functions.

Then the Crouzeix–Raviart MsFEM has been generalized for solving the highly oscillatory elliptic problem [103], diffusion problem [102], advection–diffusion problem [54, 104, 105, 113], Stokes problem [95, 118] and Oseen problem [117] in perforated domains. In order to use simple Cartesian meshes instead of boundary-fitted unstructured meshes, the penalization technique [21] is applied in the Crouzeix–Raviart MsFEM so that the velocity is forced to vanish on the boundaries of obstacles. More recently, [50] proposes an interesting multiscale hydrid high-order method for highly oscillatory elliptic problems. This method can be considered as a generalization of the Crouzeix–Raviart MsFEM to arbitrary orders of approximation. Besides, the authors provide an energy-error estimate in the case of periodic coefficients.

1.4.2.2 Cost and performance of multiscale finite element methods

The steps and advantages of multiscale method are illustrated in Figure 1.5. In multiscale simulations, a large amount of computing time comes from the construction of multiscale basis functions. However, these basis functions can be computed in parallel on a massively parallel computer so that the computing time of basis functions is reduced significantly. In practice, multiscale method allows great computational savings because basis functions are computed only once and can be re-used multiple times, for different external parameters such as source terms or boundary conditions. Significant computational savings can also be obtained for time-dependent problems in which the heterogeneities representing porous media properties do not change with time. For these problems, basis functions are pre-computed at the initial time and used throughout the simulations. The work of [131] proposes a multiscale finite element method in which time-dependent basis functions are computed for transient problem.

We resume advantages of multiscale finite element methods [87]: (1) construction of multiscale basis function in parallel; (2) re-use of multiscale basis functions for different external parameters and boundary conditions; (3) inexpensive and efficient solve of coarse-scale problems; (4) adaptive reconstruction of fine-scale features of the solution in the regions of interest.

1.5 Main contributions of the thesis

This thesis is devoted to the development of nonconforming Crouzeix-Raviart multiscale finite element methods to solve incompressible flow problems in heterogeneous media. We present the two primary objectives of this thesis in what follows.

The first purpose of this thesis is to develop a kind of model reduction technique which permits to solve flow problems in highly heterogeneous media with desired accuracy and at reasonable computational costs. For application and efficiency purpose, this technique is implemented and validated in the internal software *TrioCFD* [139] of CEA, which focus on the finite volume element method. We choose to develop nonconforming Crouzeix-Raviart multiscale finite element methods based on the recent work of [95, 117, 118]. In this thesis, the Crouzeix-Raviart multiscale finite element method for Stokes equations proposed in [95, 118] are extended to solve also Oseen problems and Navier-Stokes problems at different Reynolds numbers on the coarse mesh. The *Streamline Upwind/Petrov-Galerkin* (SUPG) method [33] is developed in the multiscale context to stabilize the solution on the coarse mesh. Moreover, we redefine the Crouzeix-Raviart multiscale finite element method proposed for Oseen equations in [117]. We provide a proof of the well-posedness of the newly-defined local problems. This new definition precludes oscillations in the solution of local problems for very large values of Oseen velocity. This multiscale method is also extended to solve Navier-Stokes problems at different Reynolds numbers on the coarse mesh with the SUPG stabilization technique. Besides, our implementation of multiscale finite element methods is original since it is adapted to existing numerical methods and solvers in *TrioCFD*, which is quite different from the implementation in [95, 117, 118]. For the first time, we extend the Crouzeix-Raviart multiscale methods to three-dimensional heterogeneous media and demonstrate the performance of these methods.

The second goal of the thesis is to improve the accuracy of existing Crouzeix-Raviart multiscale finite element methods for flow problems in heterogeneous media. This is achieved by enriching the multiscale approximation space using two types of enrichment methods.

The first type of enrichment method enriches only the approximation space of velocity and we propose three ideas for this. Firstly, the authors of [95] propose to enrich the approximation space of velocity by adding weighting functions which are defined by linear polynomials. Inspired by this idea, we propose to add more weighting functions which are defined by linear and higher-degree polynomials. Numerical experiments show that this enriched multiscale method can effectively improve the accuracy of numerical results to some degree. However, adding more than a certain number of weighting functions cannot further improve the accuracy of method. Secondly, we construct an enriched approximation space of velocity by taking the union of solutions of local Stokes problems and Oseen problems. Contrary to expectations, numerical tests show that this enriched multiscale method does not outperform the multiscale method defined by Stokes equations alone or Oseen equations alone. Thirdly, the authors of [54, 102, 113] defined bubble functions for diffusion and advection-diffusion problems to improve the accuracy of Crouzeix-Raviart multiscale finite element methods. In this thesis, we propose to define bubble functions for Stokes equations or Oseen equations. Then our numerical experiments showed that the addition of bubble functions did not improve the accuracy of the method. We thus performed theoretical analysis which proved that the approximation spaces must be altered and this led to the second type of enrichment method presented below.

The second type of enrichment method is an original generalization of the method proposed in [95]. Both the approximation space of velocity and of pressure are enriched by adding weighting functions which are defined by polynomials of different degrees. We can vary the degrees of these polynomials to obtain a compromise between the computational cost and the desired accuracy. Thus this enrichment method provides a much more general definition of the approximation space compared to that of [95, 117, 118]. We name this innovative multiscale method as the high-order Crouzeix-Raviart multiscale method. We present in detail the construction of high-order

Crouzeix-Raviart multiscale methods and define local problems respectively by Stokes equations and Oseen equations. Moreover, we point out that high-order multiscale methods would be more accurate when local problems are discretized by higher-order finite elements such as the $\mathbb{P}_2/\mathbb{P}_1$ finite element. Numerical experiments reveal that high-ordre multiscale methods can effectively improve the accuracy of both velocity and pressure solutions.

At last, we build a multiscale simulation chain mainly based on *SALOME* [129] and *TrioCFD*. Necessary meshes are generated in *SALOME* while multiscale finite element methods are implemented in *TrioCFD*. Numerical experiments are performed to study the numerical convergencce and compare the accuracy of Crouzeix-Raviart multiscale finite element methods. We present some demonstrative two- and three-dimensional applications to show the superior performance of multiscale finite element methods compared to conventional numeical methods in heterogeneous media with many solid obstacles.

1.6 Contents of the thesis

This thesis is divided into seven chapters. The present chapter, Chapter 1, is an introduction to the topic as well as the motivation of the thesis.

Chapter 2 deals with the theoretical framework of multiscale finite element methods. We first recall classical theories of numerical analysis which guarantee the existence and the uniqueness of the solution to flow problems. Afterwards, we present the Crouzeix-Raviart finite element and the finite element formulation of Stokes problems as well as its well-posedness. After these preliminary preparations, we introduce the original multiscale finite element method proposed by [87] and present each step in detail. At last, we discuss briefly the computational cost of multiscale finite element methods compared to conventional numerical methods.

Chapter 3 addresses several Crouzeix-Raviart multiscale finite element methods to solve flow problems in heterogeneous media. The chapter begins by reviewing the main idea of Crouzeix-Raviart multiscale method as well as basic definitions and notations. Then we present the Crouzeix-Raviart multiscale method originally proposed by [95, 118] where local problems are defined by Stokes equations. We propose to solve Oseen problems and Navier-Stokes problems on the coarse mesh using this method. The SUPG stabilization technique is developed in the multiscale context to stabilize the solution on the coarse mesh. Then we redefine the Crouzeix-Raviart multiscale method originally proposed by [117] where local problems are defined by Oseen equations. We provide a proof of the well-posedness of newly-defined local problems. This new definition precludes oscillations in the solution of local problems for large values of Oseen velocity. Similarly, we propose to solve Navier-Stokes problems on the coarse mesh along with the SUPG stabilization technique.

The remaining part of this chapter seeks primarily to improve the accuracy of the Crouzeix-Raviart multiscale methods defined by Stokes equations or Oseen equations. For this purpose, we investigate two ideas to enrich the approximation space. The first idea is to enrich only the the approximation space of velocity (1) by adding more weighting functions which are defined by linear and higher-degree polynomials; (2) by taking the union of solutions of local Stokes problems and Oseen problems; (3) by adding bubble functions. Our numerical experiments showed that the addition of bubble functions did not improve the accuracy of the method. We thus performed theoretical analysis which proved that the approximation spaces must be altered and this led to the second idea of enrichment below.

The second idea is to enrich both the approximation space of velocity and of pressure by adding weighting functions which are defined by polynomials of different degrees. The degrees of polynomials can vary to obtain a compromise between the computational cost and the desired accuracy. Thus this enrichment method provides a much more general definition of the approximation space compared to that of [95, 117, 118]. We name this multiscale method as the high-order Crouzeix-Raviart multiscale method. We present in detail the construction of high-order multiscale methods and define local problems respectively by Stokes equations and Oseen equations.

Chapter 4 focus on the technical aspects of the implementation of Crouzeix-Raviart multiscale finite element methods in *TrioCFD*. We start by introducing the finite volume element method and the Crouzeix-Raviart finite element. Then we present the discretization of local Stokes problem by the finite volume element method where unknowns are discretized by the Crouzeix-Raviart finite element. For local Oseen problems, we present the discretization of the additional Oseen term using the Upwind scheme. Then local problems are solved by modifying the original *prediction-correction* algorithm in *TrioCFD*. Afterwards, we present the discretization of the coarse-scale problem by the Galerkin method where basis functions are solutions of local problems. We compute matrices locally on coarse elements by taking advantage of the Crouzeix-Raviart finite element basis functions. The coarse-scale problem is also solved by the *prediction-correction* algorithm.

The remaining part of this chapter presents the discretization of local problems in high-order Crouzeix-Raviart multiscale methods by the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element. The coarse-scale problem is discretized by the Galerkin method where basis functions are solutions of local problems. Then we present the solution of both local and coarse-scale problems by a direct solver developed in this thesis. At last, we compare solutions of local problems computed by *TrioCFD* with those computed by *Freefem++* in order to validate our implementation.

Chapter 5 is devoted to practical implementations of the multiscale simulation chain *SALOME-TrioCFD-VisIt*. We present two types of parallelism: the extra-cellular parallelism and intra-cellular parallelism. We present the main steps of the multiscale simulation chain as well as parallelisms in each step. Then we introduce the *SALOME* platform and present three algorithms developed in the thesis for generating respectively conforming meshes, nonconforming meshes or meshes in periodic heterogeneous media. Then we present briefly the software *TrioCFD* and important implementations related to Crouzeix-Raviart multiscale finite element methods. In the end, we introduce briefly *VisIt* and the visualization of reconstructed fine-scale solutions.

Chapter 6 presents two- and three-dimensional numerical simulations performed with the multiscale simulation chain *SALOME-TrioCFD-VisIt*. We consider two types of media: the non-periodic heterogeneous media with randomly placed obstacles and the periodic heterogeneous media with periodically placed obstacles.

In two-dimensional non-periodic heterogeneous media, we present numerical convergence studies and error analysis of multiscale methods presented in Chapter 3. Our objective is to show the numerical convergence of multiscale methods and to compare the accuracy of enriched multiscale methods. We find that the addition of bubble functions does not improve the accuracy of the method which is consistent with our theoretical analysis. Contrary to expectations, the multiscale method enriched by adding solutions of local Stokes problems and Oseen problems does not outperform the multiscale method defined by Stokes equations alone or Oseen equations alone. Interestingly, numerical experiments show that high-order multiscale methods can significantly improve the accuracy of both the velocity and pressure. Moreover, we solve the nonlinear Navier-Stokes problems at different Reynolds numbers with Crouzeix-Raviart multiscale methods defined by Stokes or Oseen equations.

In two-dimensional periodic heterogeneous media, our numerical convergence studies show that it is necessary to enrich multiscale methods in order to obtain rather accurate solutions. Besides, we generate coarse meshes using two partition methods and compare the accuracy of multiscale methods using these partitions. At last, we present several applications of Crouzeix-Raviart multiscale methods in two- and three-dimensional highly heterogeneous media to demonstrate the performance of multiscale methods.

The last chapter concludes this thesis by giving an overview of Crouzeix-Raviart multiscale finite elements developed in the thesis and discussing the performance of these multiscale methods. We review some important remarks made during the thesis and propose some ideas and perspectives to improve the performance of the multiscale simulation chain.

Chapter 2

Theoretical framework of multiscale finite element methods

This chapter deals with the theoretical framework of multiscale finite element methods. Before introducing multiscale finite element methods, we first recall some classical theories of numerical analysis. These theories guarantee the existence and uniqueness of a solution to flow problems, in particular, Stokes problems and Oseen problems. Meanwhile, we recall these flow problems and define their variational formulations. The existence and uniqueness of a solution to these problems are guaranteed by previously presented theories. Afterwards, we present the Crouzeix-Raviart finite element and the finite element formulations of flows problems as well as their well-posedness. After these preliminary preparations, we finally introduce the main idea of the original multiscale finite element methods proposed by [87]. This introduction makes it easier to understand Crouzeix-Raviart multiscale finite element methods which will be presented in the next chapter. At last, the cost of multiscale finite element methods and traditional finite element methods are compared in order to show the good performance of the first one.

Outline Sections 2.1 to 2.3 introduces the theories of numerical analysis of an abstract problem and variational formulations of flow problems. Section 2.4 describes briefly the finite element method and the Crouzeix-Raviart finite element. Section 2.5 addresses the classical multiscale finite element method. Section 2.6 evaluates the the cost and performance of multiscale finite element methods.

2.1 Analysis of an abstract variational problem

This section recalls briefly some classical theories of numerical analysis of an abstract problem. For more details about these theories, the reader can refer to many books on finite element methods, such as [67, 73].

Let V and M be two Hilbert spaces [9]. The scalar product defined on these spaces are denoted respectively by $(\cdot,\cdot)_V$ and $(\cdot,\cdot)_M$. The norms associated to these scalar products are denoted respectively by $\|\cdot\|_V$ and $\|\cdot\|_M$. Let V' and M' be the dual spaces of V and M and let $\|\cdot\|_{V'}$ and $\|\cdot\|_{M'}$ be the associated dual norms. The dual space V' (respectively M') is the space of linear forms defined on V (respectively M). We denote by $\langle\cdot,\cdot\rangle$ the product of an element in the Hilbert space and an element of its dual space. Let f and g be element of V' and M' respectively, i.e. f and g are two linear forms.

Let $a(.,.)$ and $b(.,.)$ be two continuous bilinear forms:

$$a(.,.) : V \times V \to \mathbb{R}, \quad b(.,.) : V \times M \to \mathbb{R}$$

Then we consider the following variational problem: Given $f \in V'$ and $g \in M'$, find $(u, p) \in V \times M$ such that

$$\begin{cases} a\left(u,v\right) + b\left(v,p\right) = \langle f,v\rangle \quad \forall v \in V \\ b\left(u,q\right) = \langle g,q\rangle \quad \forall q \in M \end{cases} \tag{2.1}$$

Our purpose is to derive the necessary and sufficient conditions so that the problem (2.1) is well-posed (the problem has one and only one solution). Let W be a subspace of V and W is defined by:

$$W = \{v \in V |\, \forall q \in M,\ b\left(v,q\right) = 0\}$$

We suppose firstly that $g = 0$. The initial problem (2.1) can be rewritten as: Find $u \in W$ such that

$$a\left(u,v\right) = \langle f,v\rangle \quad \forall v \in W \tag{2.2}$$

The well-posedness of problem (2.2) is ensured by the following theorem (see theorem 1.7. of [73]) which is due to Lax & Milgram's theorem [101].

Theorem 2.1.1. *We assume that*
– $a(.,.)$ *is continuous, i.e., there exists a constant* β *such that*

$$|a\left(u,v\right)| \leq \beta \left\|u\right\|_W \left\|v\right\|_W \quad \forall u,v \in W$$

– $a(.,.)$ *is elliptic on* W*, i.e., there exists a constant* α *such that*

$$a\left(v,v\right) \geq \alpha \left\|v\right\|_W^2 \quad \forall v \in W$$

Then the problem (2.2) has one and only one solution $u \in W$*. Moreover, the mapping* $f \to u$ *is an isomorphism from* W' *onto* W*.*

The work of [67] proposes a more general theorem than Lax & Milgram's theorem for the bilinear form $a(.,.)$. We recall the theorem below:

Theorem 2.1.2 (Banach-Nečas-Babuška (BNB)). *Let* X *be a Banach space and let* Y *be a reflexive Banach space. Let* a *be a bilinear form:* $X \times Y \to \mathbb{R}$ *and* $f \in Y'$*. Then the problem (2.2) is well-posed if and only if:*

$$\exists \alpha > 0, \quad \inf_{w\in X} \sup_{v \in Y} \frac{a\left(w,v\right)}{\left\|w\right\|_X \left\|v\right\|_Y} \geq \alpha$$
$$\forall v \in Y, \quad (\forall w \in X, a(w,v) = 0) \Rightarrow (v = 0)$$

Moreover, the following a priori estimate holds:

$$\forall f \in Y', \quad \left\|u\right\|_X \leq \frac{1}{\alpha} \left\|f\right\|_{Y'}$$

This theorem is proved in [67] and it is stated that Theorem 2.1.1 is a consequence of Theorem 2.1.2. We will apply this theorem directly in what follows.

Now we need to consider how to treat the case $g \neq 0$ and discuss the existence and uniqueness of p to problem (2.1). To do this, we introduce the inf-sup condition introduced by Babuška-Brezzi [28, 32].

Theorem 2.1.3 (Babuška-Brezzi). *The three following properties are equivalent:*
1, there exists a constant $\beta > 0$ *such that*

$$\inf_{q \in M, q \neq 0} \sup_{v \in V} \frac{b(v,q)}{\|v\|_V \|q\|_M} \geq \beta$$

2, there exists an isomorphism B' *from* M *onto* W° *which verifies:*

$$b(v,q) = \langle v;\ B'q \rangle_{V,V'}, \quad \|B'q\|_{V'} \geq \beta \|q\|_M \ \forall q \in M$$

The space W° *is defined as:* $W^{\circ} = \{h \in V' | \ \forall v \in W, \langle h, v \rangle = 0\}$.
3, there exists an isomorphism B *from* $W^{\perp}$ *onto* M' *which verifies that*

$$b(v,q) = \langle Bv;\ q \rangle_{M',M}, \quad \|Bv\|_{M'} \geq \beta \|v\|_V \ \forall v \in W^{\perp}$$

The orthogonal space $W^{\perp}$ *of* W *is defined as* $W^{\perp} = \{v \in V | \ \forall w \in W, (v,w) = 0\}$.

This theorem is proved in [73] and we apply it directly in what follows.

Theorem 2.1.4. *Assume that*
1, $a(.,.)$ *is a bilinear form continuous on* $V \times V$.
2, $b(.,.)$ *is a bilinear form continuous on* $V \times M$.
3, $a(.,.)$ *is* V*–elliptic, i.e. there exists a constant* $\alpha > 0$ *such that*

$$a(v,v) \geq \alpha \|v\|_V^2 \quad \forall v \in V$$

4, $b(.,.)$ *verifies the condition inf-sup: there exists a constant* $\beta > 0$ *such that*

$$\inf_{q \in M, q \neq 0} \sup_{v \in V} \frac{b(v,q)}{\|v\|_V \|q\|_M} \geq \beta$$

Then problem (2.1) is well-posed and it has one unique solution $u \in V,\ p \in M$ *for any* $f \in V',\ g \in M'$.

This theorem can be proved easily using Theorem 2.1.3 and we apply directly this theorem in what follows.

2.2 The variational formulation of Stokes problem

Let Ω be a connected and bounded open set in $\mathbb{R}^d$, $d = 2$ or 3, with a Lipschitz-continous boundary. The Stokes problem with homogeneous Dirichlet boundary condition is: find the velocity $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ and the pressure $p : \Omega \to \mathbb{R}$ solutions to:

$$-\mu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \text{ in } \Omega \tag{2.3}$$

$$\text{div } \boldsymbol{u} = g \text{ in } \Omega \tag{2.4}$$

$$\boldsymbol{u} = \mathbf{0} \text{ on } \partial\Omega$$

where μ is the dynamic viscosity, $\boldsymbol{f}$ is a given force and g is a given function. In particular, $g = 0$ for incompressible flows.

We introduce the following Sobolev spaces [9]:

- $L^2(\Omega)$ is the space of square integrable functions.
- $L_0^2(\Omega) = \left\{p \in L^2(\Omega) \,\middle|\, \int_\Omega p = 0\right\}$ is a subspace of $L^2(\Omega)$

- $\left(H^1(\Omega)\right)^d = \left\{ \boldsymbol{v} \in \left(L^2\left(\Omega\right)\right)^d \mid \nabla \boldsymbol{v} \in \left(L^2\left(\Omega\right)\right)^d \right\}$
- $\left(H_0^1(\Omega)\right)^d$ is a subspace of $\left(H^1(\Omega)\right)^d$ and it contains functions whose trace is zero on $\partial\Omega$

The scalar product of $L^2(\Omega)$ and the associated norm are denoted respectively by $(.,.)$ and by $\|\cdot\|_{L^2}$.

$$(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{u} \cdot \boldsymbol{v}, \quad \|\boldsymbol{u}\|_{L^2} = (\boldsymbol{u}, \boldsymbol{u})^{\frac{1}{2}}$$

The norm for the space $\left(H_0^1(\Omega)\right)^d$ is defined as

$$\|\boldsymbol{u}\|_0 = \left(\int_\Omega |\nabla \boldsymbol{u}|^2 \right)^{1/2}$$

We introduce the following bilinear forms:

$$a^{St}(\boldsymbol{u}, \boldsymbol{v}) = \mu \int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{v}, \quad b(\boldsymbol{v}, q) = -\int_\Omega q \text{ div } \boldsymbol{v}$$

and linear forms:

$$F\left(\boldsymbol{v}\right) = \langle \boldsymbol{f}, \boldsymbol{v} \rangle = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}, \quad G\left(q\right) = \langle g, q \rangle = -\int_\Omega gq \quad \forall \boldsymbol{v}_h \in V_h, \ \forall q_h \in M_h$$

Now we try to write the variational formulation of Stokes equations (2.3)–(2.4). Multiplying (2.3) by a test function $\boldsymbol{v} \in \left(H_0^1(\Omega)\right)^d$ and integrating over Ω, we obtain

$$\int_\Omega \mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} - \int_\Omega p \text{ div} \boldsymbol{v} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}$$

Similarly, multiplying (2.4) by a function $q \in L^2\left(\Omega\right)$ and integrating on the domain Ω, we have

$$\int_\Omega \text{div} \boldsymbol{u} \; q = \int_\Omega gq$$

With the notations above, the variational formulation of Stokes equations (2.3)–(2.4) can be written in the abstract form: find $\boldsymbol{u} \in \left(H_0^1\left(\Omega\right)\right)^d$ and $p \in L_0^2\left(\Omega\right)$ such that

$$\begin{cases} a^{St}(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = F\left(\boldsymbol{v}\right) & \forall \boldsymbol{v} \in \left(H_0^1\left(\Omega\right)\right)^d \\ b(\boldsymbol{u}, q) = G\left(q\right) & \forall q \in L_0^2\left(\Omega\right) \end{cases} \tag{2.5}$$

In conclusion, the Stokes problem is one prototype example of problem (2.1), by choosing $V = \left(H_0^1\left(\Omega\right)\right)^d$ and $M = L_0^2\left(\Omega\right)$. The space W reads:

$$W = \left\{ \boldsymbol{v} \in \left(H_0^1\left(\Omega\right)\right)^d \mid \forall q \in L_0^2\left(\Omega\right), b(\boldsymbol{v}, q) = 0 \right\} = \left\{ \boldsymbol{v} \in \left(H_0^1\left(\Omega\right)\right)^d \mid \text{div } \boldsymbol{v} = 0 \right\}$$

The existence and uniqueness of a solution to system (2.5) is guaranteed by Theorem 2.1.4. We see that each of the hypothesis of Theorem 2.1.4 is verified:

- bilinear forms $a^{St}(.,.)$ and $b(.,.)$ are continuous respectively on $\left(H_0^1\left(\Omega\right)\right)^d \times \left(H_0^1\left(\Omega\right)\right)^d$ and $\left(H_0^1\left(\Omega\right)\right)^d \times L_0^2\left(\Omega\right)$.

- $a^{St}(.,.)$ is elliptic on V. We recall the Poincaré inequality: there exists a $C > 0$ such that

$$\forall \boldsymbol{v} \in \left(H_0^1(\Omega)\right)^d, \; a^{St}(\boldsymbol{v}, \boldsymbol{v}) = \int_\Omega \mu |\nabla \boldsymbol{v}|^2 \geq C \int_\Omega \mu |\boldsymbol{v}|^2 \tag{2.6}$$

  Since the semi-norm $\|\nabla \boldsymbol{v}\|_{L^2(\Omega)}$ is equivalent to the full H^1 norm by the Poincaré inequality, we obtain easily that $a^{St}(.,.)$ is elliptic on V.

- the bilinear form $b(.,.)$ verifies the inf-sup condition. Since the gradient operator is an isomorphism from $V^\perp$ onto $\frac{L^2(\Omega)}{\mathbb{R}}$ and the divergence operator is an isomorphism from $L_0^2(\Omega)$ onto V°, Theorem 2.1.3 implies that $b(.,.)$ verifies the inf-sup condition of Theorem 2.1.4.

We introduce another bilinear form $c^{St}(.,.)$ continuous on $(V \times M)^2$:

$$c^{St}((\boldsymbol{u}, p), (\boldsymbol{v}, q)) = \int_\Omega \mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} - \int_\Omega p \text{ div } \boldsymbol{v} - \int_\Omega q \text{ div } \boldsymbol{u} \tag{2.7}$$

The variational formulation of (2.3)–(2.4) is equivalent to: find $(\boldsymbol{u}, p) \in V \times M$ such that

$$c^{St}((\boldsymbol{u}, p), (\boldsymbol{v}, q)) = F(\boldsymbol{v}) \quad \forall (\boldsymbol{v}, q) \in V \times M \tag{2.8}$$

Let $X = V \times M$ with $V = \left(H_0^1(\Omega)\right)^d$ and $M = L_0^2(\Omega)$. Theorem 2.1.2 implies that problem (2.8) has a unique solution if the bilinear form $c^{St}(.,.)$ satisfies the following inf-sup property:

$$\inf_{(\boldsymbol{u},p)\in X} \sup_{(\boldsymbol{v},q)\in X} \frac{c^{St}\left((\boldsymbol{u}, p), (\boldsymbol{v}, q)\right)}{\|\boldsymbol{u}, p\|_X \|\boldsymbol{v}, q\|_X} \geq \gamma \tag{2.9}$$

with a constant $\gamma > 0$.

2.3 The variational formulation of Oseen problem

Let Ω be a connected and bounded open set in $\mathbb{R}^d$, $d = 2$ or 3, with a Lipschitz-continous boundary. The steady-state Oseen problem with homogeneous Dirichlet boundary condition is to find the velocity $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ and the pressure $p : \Omega \to \mathbb{R}$ solutions to:

$$-\mu \Delta \boldsymbol{u} + \rho \left(\boldsymbol{U}_o \cdot \nabla\right) \boldsymbol{u} + \nabla p = \boldsymbol{f} \text{ in } \Omega \tag{2.10}$$

$$\text{div } \boldsymbol{u} = g \text{ in } \Omega \tag{2.11}$$

$$\boldsymbol{u} = \boldsymbol{0} \text{ on } \partial\Omega \tag{2.12}$$

where μ is the dynamic viscosity, $\boldsymbol{U}_o$ is a known velocity, ρ is the flow density, $\boldsymbol{f}$ is a given force and g is a given function. In particular, $g = 0$ for incompressible flows.

We introduce the bilinear form $a^{Os}(.,.)$ for the Oseen problem:

$$a^{Os}(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \left(\mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \rho \left(\boldsymbol{U}_o \cdot \nabla\right) \boldsymbol{u} \cdot \boldsymbol{v}\right) \tag{2.13}$$

Thus the variational formulation of the Oseen problem is to find $\boldsymbol{u} \in \left(H_0^1(\Omega)\right)^d$ and $p \in L_0^2(\Omega)$ such that

$$\begin{cases} a^{Os}(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = F(\boldsymbol{v}) & \forall \boldsymbol{v} \in \left(H_0^1(\Omega)\right)^d \\ b(\boldsymbol{u}, q) = G(q) & \forall q \in L_0^2(\Omega) \end{cases} \tag{2.14}$$

By choosing $V = \left(H_0^1(\Omega)\right)^d$ and $M = L_0^2(\Omega)$, we apply Theorem 2.1.4 to guarantee the existence and uniqueness of a solution to (2.14). It is clear that the hypothesis 1, 2 and 4 of Theorem 2.1.4 are verified by bilinear forms $a^{Os}(.,.)$ and $b(.,.)$.

We introduce another continuous bilinear form which is equivalent to $a^{Os}(.,.)$:

$$\widehat{a}^{Os}(\boldsymbol{u},\boldsymbol{v}) = \int_\Omega \left(\mu\nabla\boldsymbol{u} : \nabla\boldsymbol{v} + \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{u}\right)\boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right)\boldsymbol{u} - \frac{1}{2}\rho\boldsymbol{u}\boldsymbol{v}\text{ div }\boldsymbol{U}_o\right)$$

with $\boldsymbol{u} = \boldsymbol{0}$ on $\partial\Omega$ (boundary condition (2.12)).

Since the two bilinear forms $a^{Os}(.,.)$ and $\widehat{a}^{Os}(.,.)$ are equivalent and it is easier to prove that the hypothesis 3 holds using $\widehat{a}^{Os}(.,.)$, we now prove that $\widehat{a}^{Os}$ is coercive on V. It is easy to observe that

$$\begin{aligned}\widehat{a}^{Os}(\boldsymbol{u},\boldsymbol{u}) &= \int_\Omega \left(\mu\nabla\boldsymbol{u} : \nabla\boldsymbol{u} - \frac{1}{2}\rho\boldsymbol{u}\boldsymbol{u}\text{ div }\boldsymbol{U}_o\right)\\ &\geq \int_\Omega \mu|\nabla\boldsymbol{u}|^2\end{aligned}$$

by assuming that div $\boldsymbol{U}_o \leq 0$.

Since the semi-norm $\|\nabla\boldsymbol{v}\|_{L^2(\Omega)}$ is equivalent to the full H^1 norm by Poincaré inequality (2.6), we obtain that $a^{St}(.,.)$ is elliptic on V. Thus the hypothesis 3 holds for $\widehat{a}^{Os}(.,.)$ and $a^{Os}(.,.)$.

Consequently Theorem 2.1.4 guarantees that the following variational formulation of Oseen problem is well-posed.

$$\begin{cases}\widehat{a}^{Os}(\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) = F(\boldsymbol{v}) & \forall\boldsymbol{v}\in\left(H_0^1(\Omega)\right)^d\\ b(\boldsymbol{u},q) = G(q) \quad \forall q\in L_0^2(\Omega)\end{cases} \tag{2.15}$$

We introduce also the bilinear form $\widehat{c}^{Os}(.,.)$:

$$\begin{aligned}\widehat{c}^{Os}((\boldsymbol{u},p),(\boldsymbol{v},q)) = \int_\Omega &\left(\mu\nabla\boldsymbol{u} : \nabla\boldsymbol{v} + \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{u}\right)\boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right)\boldsymbol{u}\right)\\ &- \int_\Omega p\text{ div }\boldsymbol{v} - \int_\Omega q\text{ div }\boldsymbol{u}\end{aligned}$$

The variational formula of the Oseen problem can be written as: find $(\boldsymbol{u},p)\in V\times M$ such that

$$\widehat{c}^{Os}((\boldsymbol{u},p),(\boldsymbol{v},q)) = F(\boldsymbol{v}) \quad \forall\,(\boldsymbol{v},q)\in V\times M$$

By choosing $X = V\times M$, Theorem 2.1.2 guarantees the existence and uniqueness of a solution to this problem.

2.3.1 The variational formulation of Navier-Stokes problem

Let Ω be a connected and bounded open set in $\mathbb{R}^d$, $d = 2$ or 3, with a Lipschitz-continous boundary. The steady-state Navier-Stokes problem with homogeneous Dirichlet boundary condition is to find the velocity $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ and the pressure $p : \Omega\to\mathbb{R}$ solutions to:

$$-\mu\Delta\boldsymbol{u} + \rho\left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u} + \nabla p = \boldsymbol{f}\text{ in }\Omega \tag{2.16}$$

$$\text{div }\boldsymbol{u} = g\text{ in }\Omega \tag{2.17}$$

$$\boldsymbol{u} = \boldsymbol{0}\text{ on }\partial\Omega$$

where μ is the dynamic viscosity and ρ is the flow density, $\boldsymbol{f}$ is a given force and g is a given function. In particular, $g = 0$ for incompressible flows.

We introduce the following nonlinear forms:

$$\begin{aligned} a^{NS}(\boldsymbol{u}, \boldsymbol{v}) &= \int_\Omega (\mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \rho (\boldsymbol{u} \cdot \nabla) \boldsymbol{u} \cdot \boldsymbol{v}) \\ c^{NS}((\boldsymbol{u}, p), (\boldsymbol{v}, q)) &= \int_\Omega (\mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \rho (\boldsymbol{u} \cdot \nabla) \boldsymbol{u} \cdot \boldsymbol{v}) - \int_\Omega p \operatorname{div} \boldsymbol{v} - \int_\Omega q \operatorname{div} \boldsymbol{u} \end{aligned}$$

The generalization of the abstract variational problem analyzed in section 2.1 to nonlinear problems can be found in [73]. The family of nonlinear problems contains the Navier-Stokes problem in particular. The analysis of a nonlinear abstract variational problem is much more complicated than a linear abstract problem. Thus we choose not to present the numerical analysis of Navier-Stokes problem in this thesis.

2.4 The finite element method

The finite element method is a very popular method to solve Partial Differential Equations (PDEs). It approximates the continuous space of solution of PDEs by a finite dimensional space. To show the main idea of the method, we present the approximation of the abstract variational problem analyzed in section 2.1.

Let h denote a discretization parameter tending to zero. For each h, let V_h and M_h two finite dimensional spaces that V_h and M_h. We introduce two bilinear forms $a_h(.,.)$ and $b_h(.,.)$ on $V_h \times V_h$ and $V_h \times M_h$ respectively. We now approximate the problem (2.1) by the discrete problem: Given $f \in V'$ and $g \in M'$, find $(u_h, p_h) \in V_h \times M_h$ such that

$$\begin{cases} a_h(u_h, v_h) + b_h(v_h, p_h) = \langle f, v_h \rangle & \forall v_h \in V_h \\ b_h(u_h, q_h) = \langle g, q_h \rangle & \forall q \in M_h \end{cases} \tag{2.18}$$

The existence and uniqueness of a solution to system (2.18) is guaranteed by the following discrete inf-sup condition Theorem 2.4.1, which is a discrete version of Theorem 2.1.4:

Theorem 2.4.1. *Assume that*
1, $a_h(.,.)$ is a bilinear form continuous on $V_h \times V_h$.
2, $b_h(.,.)$ is a bilinear form continuous on $V_h \times M_h$.
3, $a_h(.,.)$ is V_h-elliptic, i.e. there exists a constant $\alpha_h > 0$ such that

$$a_h(v_h, v_h) \geq \alpha_h \|v\|_{V_h}^2 \quad \forall v_h \in V_h$$

4, $b_h(.,.)$ verifies inf-sup condition: there exists a constant $\beta_h > 0$ such that

$$\inf_{q_h \in M_h, q_h \neq 0} \sup_{v_h \in V_h} \frac{b_h(v_h, q_h)}{\|v_h\|_{V_h} \|q_h\|_{M_h}} \geq \beta_h$$

Then problem (2.18) is well-posed and it has one unique solution $u_h \in V_h$, $p_h \in M_h$ for any $f \in V'$, $g \in M'$.

2.4.1 The Crouzeix-Raviart finite element

Let Ω be a connected and bounded open set in $\mathbb{R}^d$ with $d = 2$ or 3. We denote $\mathcal{T}_h$ a discretization of Ω by triangles ($d = 2$) or tetrahedrons ($d = 3$) noted as K. The i-th face of $\mathcal{T}_h$ is noted as f_i and the middle of the face f_i is noted as x_i. The barycenter of the triangle or tetrahedron is noted as g_K. The set of faces in the discretization is noted as $\mathcal{E}_h$. The Crouzeix-Raviart finite element

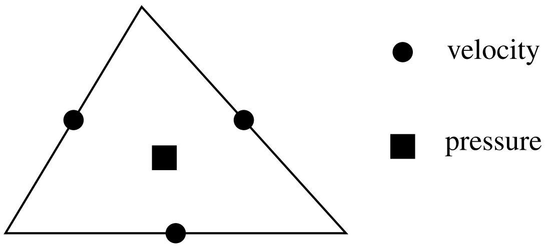


FIGURE 2.1: Crouzeix-Raviart element with locations of unknowns ($d = 2$)

shown in Figure 2.1 was first introduced in [52]. The velocity unknown of this element is in the barycenter of each face and the pressure unknown is in the barycenter of each element.

We note V_h and M_h as the approximation space of the velocity u_h and the pressure p_h respectively. The spaces V_h and M_h are defined by

$$V_h = \left\{ \boldsymbol{v}_h \in \left(L^2\left(\Omega\right)\right)^d \mid \boldsymbol{v}_h|_K \in \left(\mathbb{P}_1(K)\right)^d,\ \boldsymbol{v}_h \text{ is continuous at points } x_i,\ \forall K \in \mathcal{T}_h \right\}$$
$$M_h = \left\{ q_h \in L_0^2\left(\Omega\right) \mid q_h|_K \in \mathbb{P}_0(K),\ \forall K \in \mathcal{T}_h \right\}$$

The approximation space V_h is not included in $V = \left(H_0^1(\Omega)\right)^d$ and it is nonconforming. But the approximation space M_h is included in the continuous space $M = L_0^2\left(\Omega\right)$.

Let $\{a_0, \cdots, a_d\}$ be the vertices of K, f_i be the face of K opposite to a_i and n_i be the outward normal to f_i. The associated barycentric coordinates $(\lambda_0, \cdots, \lambda_d)$ are defined by:

$$\lambda_i : x \to \lambda_i(x) = 1 - \frac{(x - a_i) \cdot n_i}{(a_j - a_i) \cdot n_i} \quad \text{for } 0 \leq i \leq d$$

where a_j is an arbitrary vertex in f_i. We have $\lambda_i\left(a_j\right) = \delta_{ij}$ for any vertex a_j of K. Besides,

$$\sum_{i=0}^{d} \lambda_i(x) = 1, \quad \sum_{i=0}^{d} \lambda_i(x)(x - a_i) = 0 \quad \text{for all } x \in \mathbb{R}^d$$

Let $\{\phi_1, \cdots, \phi_n\}$ be the basis function of each component of V_h and $\tilde{V}_h$ with n the dimension of V_h or $\tilde{V}_h$. Using the barycentric coordinates $(\lambda_0, \cdots, \lambda_d)$, ϕ_i reads on each element K:

$$\phi_i|_K = 1 - d\lambda_i|_K$$

We have $\phi_i|_K\left(x_j\right) = \delta_{ij}$. Thus each function of V_h can be written as:

$$\boldsymbol{u}_h = (u_h; v_h) = \left(\sum_i u_h(x_i)\phi_i\left(x\right); \sum_i v_h(x_i)\phi_i\left(x\right) \right) = \sum_i \boldsymbol{u}_h\left(x_i\right)\phi_i\left(x\right)$$

We denote by $[[\boldsymbol{v}_h]]$ the "jump" of $\boldsymbol{v}_h$ across an internal face with $[[\boldsymbol{v}_h]] = \boldsymbol{v}_h$ on $\partial\Omega$. [67] shows that V_h can be defined equivalently as:

$$\tilde{V}_h = \left\{ \boldsymbol{v}_h \in \left(L^2\left(\Omega\right)\right)^d \mid \boldsymbol{v}_h|_K \in \left(\mathbb{P}_1(K)\right)^d,\ \int_{f_i} [[\boldsymbol{v}_h]] = \boldsymbol{0},\ \forall K \in \mathcal{T}_h,\ \forall f_i \in \mathcal{E}_h \right\}$$

Since $\phi_i|_K \in \mathbb{P}_1\left(K\right)$, the mean-value over a face is equal to the value at the barycenter. The condition $\int_{f_i} [[\boldsymbol{v}_h]] = 0$ for $f_i \in \mathcal{E}_h$ implies that functions of $\tilde{V}_h$ are only continuous at the barycenter of the face f_i. Therefore V_h and $\tilde{V}_h$ are indeed the same space. $\tilde{V}_h$ is not included in $\left(H_0^1\left(\Omega\right)\right)^d$ and it is nonconforming.

Let ψ_K be the characteristic function of K (ψ_K equals to 1 in K and 0 otherwise), each function of M_h can be written as

$$q_h\left(x\right)=\sum_{K\in\mathcal{T}_h(\Omega)}q_h\left(g_K\right)\psi_K\left(x\right)$$

where g_K is the barycenter of K.

2.4.2 A finite element formulation of Stokes problem

For the sake of simplicity, we consider Stokes problem (2.3)–(2.4) with homogeneous Dirichlet condition. We recall that the spaces V and M are $V=\left(H_0^1\left(\Omega\right)\right)^d$ and $M=L_0^2\left(\Omega\right)$ in section 2.2. In the finite element formulation, we replace V and M by the approximation space V_h and M_h, the bilinear forms $a^{St}(,.,)$ and $b(,.,)$ by $a_h^{St}(,.,)$ and $b_h(,.,)$. Since functions of V_h is not included in $\left(H^1\left(\Omega\right)\right)^d$ but in $\Pi_{K\in\mathcal{T}_h}\left(H^1(K)\right)^d$. For Crouzeix-Raviart element, we define the bilinear forms $a_h^{St}(.,.)$ and $b_h(.,.)$ by

$$\begin{aligned}a_h^{St}(\boldsymbol{u}_h,\boldsymbol{v}_h)&=\sum_{K\in\mathcal{T}_h}\int_K\nabla\boldsymbol{u}_h:\nabla\boldsymbol{v}_h\quad\forall\boldsymbol{u}_h,\boldsymbol{v}_h\in V_h\\ b_h(\boldsymbol{u}_h,q_h)&=-\sum_{K\in\mathcal{T}_h}\int_K q_h\text{div }\boldsymbol{u}_h\quad\forall\boldsymbol{u}_h\in V_h,\forall q_h\in M_h\end{aligned}$$

and the linear forms:

$$F_h(\boldsymbol{v}_h)=\sum_{K\in\mathcal{T}_h}\int_K\boldsymbol{f}\cdot\boldsymbol{v}_h,\quad G_h(q_h)=-\sum_{K\in\mathcal{T}_h}\int_K gq_h\quad\forall\boldsymbol{v}_h\in V_h,\ \forall q_h\in M_h$$

The discrete variational formulation of the Stokes problem is: find $\boldsymbol{u}_h\in V_h$ and $p_h\in M_h$ such that

$$\begin{cases}a_h^{St}\left(\boldsymbol{u}_h,\boldsymbol{v}_h\right)+b_h\left(\boldsymbol{v}_h,p_h\right)=F_h\left(\boldsymbol{v}_h\right)\quad\forall\boldsymbol{v}_h\in V_h\\ b_h\left(\boldsymbol{u}_h,q_h\right)=G_h(q_h)\quad\forall q_h\in M_h\end{cases}\tag{2.19}$$

The existence and uniqueness of a solution to problem (2.19) can be guaranteed by Theorem 2.1.4. In fact, we can show that each of the hypothesis in this theorem holds, i.e.:

- $a_h(.,.)$ is continuous on $V_h\times V_h$ using the Cauchy-Schwarz inequality.
- $b_h(.,.)$ is continuous on $V_h\times M_h$ using the Cauchy-Schwarz inequality.
- $a_h(.,.)$ is elliptic on V_h with the help of the discrete Poincaré inequality [143].
- The inf-sup condition of $b_h(.,.)$ on $V_h\times M_h$ is proved by Crouzeix and Raviart [52].

2.5 The original multiscale finite element method

Before presenting the Crouzeix-Raviart multiscale finite element methods, we present a simple example illustrating the main concept of the original multiscale finite element method proposed in [87]. We consider a second-order elliptic problem in a domain $\Omega\in\mathbb{R}^d$ with $d\in\text{span}\left\{2,3\right\}$. The problem is to find $p:\Omega\to\mathbb{R}$ solution to

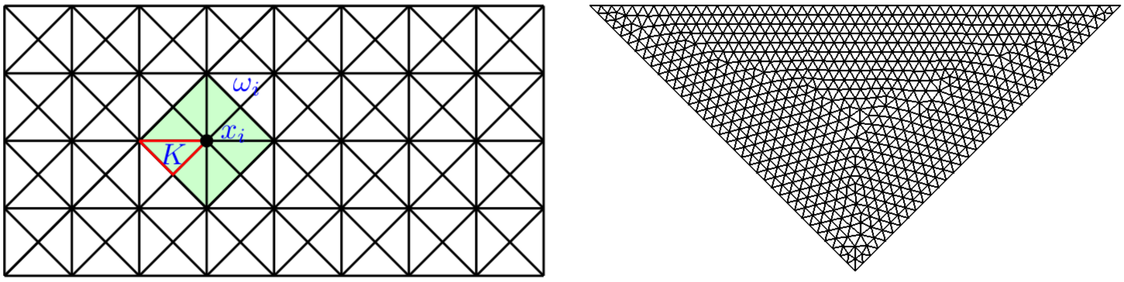
$$-\nabla\cdot a\left(\boldsymbol{x}\right)\nabla p=f\text{ in }\Omega$$

where f is a given function and $a(\boldsymbol{x}) = (a_{ij}(\boldsymbol{x}))$ is the ratio of the permeability tensor κ and the fluid viscosity μ. Besides, $a(\boldsymbol{x})$ is assumed to be symmetric and positive definite with upper and lower bounds. $a(\boldsymbol{x})$ is a heterogeneous field varying over multiple scales. The steady velocity field is related to the pressure through the Darcy's law (1.4).

The heterogeneous domain consists of three scales: the size L of the domain Ω, the coarse-scale H that corresponds to the macroscopic features of the problem and the small-scale ε that corresponds to microscopic features of the problem. The three scales verify that $\varepsilon \ll H \ll L$. We recall that multiscale finite element methods consist of two main ingredients: multiscale basis functions and a global formulation that couples the multiscale basis functions. Important multiscale features are incorporated into the locally computed basis functions. A pseudo code in Algorithm 2.1 outlines the main steps of multiscale finite element methods. We present each step in what follows.

Algorithm 2.1 Main steps of multiscale finite element methods

```
Partition the domain into a set of coarse elements (coarse mesh)      ▷ subsection 2.5.1
for each coarse element do
    Partition the element into a fine mesh
    Construct multiscale basis functions via local problems           ▷ subsection 2.5.2
    Compute matrices locally on the fine mesh
end for
Assemble global matrices and solve the coarse-scale problem           ▷ subsection 2.5.3
for coarse elements in the region of interest do
    Reconstruct fine-scale solutions on the fine mesh                 ▷ subsection 2.5.4
end for
```



(a) coarse mesh $\mathcal{T}_H$ and element K (b) coarse element $\mathcal{T}_h(K)$

FIGURE 2.2: Illustration of the coarse mesh $\mathcal{T}_H$ and one coarse element $\mathcal{T}_h(K)$

2.5.1 Discretization of the domain

The first step of multiscale finite element methods is to discretize the global domain Ω into a coarse mesh. As shown in Figure 2.2 (a), we split the domain Ω into a regular mesh $\mathcal{T}_H$ with finite elements (triangles or quadrilaterals if $d = 2$, tetrahedrons or hexagons if $d = 3$) of size at most H. The size H is presumed to be much larger than the grid size necessary for a classical finite element method to solve flow problems in the domain Ω^ε. We denote this partition as the coarse mesh (or coarse grid) and finite elements of this partition as coarse elements. For each coarse element K in $\mathcal{T}_H$, as shown in Figure 2.2 (b), we construct individually a finer mesh $\mathcal{T}_h(K)$ which consists of fine elements each with a width h and $0 < h \ll H$.

2.5.2 Computation of multiscale basis functions

As shown in Figure 2.2, let x_i be the interior nodes of the mesh $\mathcal{T}_H$ and ϕ_i^0 be the nodal basis of the standard finite element space $V_h = \text{span}\left\{\phi_i^0\right\}$. On a triangular partition, V_h consists of piecewise linear functions. Denoting the neighborhood of x_i as $\omega_i = \text{supp}\left(\phi_i^0\right)$, we define the multiscale basis function ϕ_i as solution of the following local problem:

$$\nabla \cdot a\left(\boldsymbol{x}\right)\nabla\phi_i = 0 \text{ in } K, \quad \phi_i = \phi_i^0 \text{ on } \partial K, \quad \forall K \in \mathcal{T}_H,\ K \subset \omega_i \tag{2.20}$$

In this case, the multiscale basis functions coincide with standard finite element basis functions on the boundary of coarse elements K. Since piecewise linear polynomials are imposed on the boundary ∂K, the local problem (2.20) defined on element $K \subset \omega_i$ is totally independent from that defined on the adjacent element $K' \subset \omega_i$. Thus local problems can be solved in parallel on different coarse elements of $\mathcal{T}_H$. This parallelism allows to reduce significantly the computing memory and the computing time. Moreover, the local problem (2.20) can be solved on the fine grid $\mathcal{T}_h\left(K\right)$ for $K \in \mathcal{T}_H$ by a standard finite element method or finite volume method. Then we denote by Y_H the finite-dimensional space spanned by ϕ_i

$$Y_H = \text{span}\left\{\phi_i\right\}$$

2.5.3 The coarse-scale formulation

The approximation u_H of the solution is:

$$p_H(x) = \sum_i p_H^i \phi_i(x) \quad \forall x \in \Omega \tag{2.21}$$

with p_H^i the solution value at the nodal point x_i of the coarse mesh.

In the case of Galerkin finite element methods, when the basis functions are conforming ($Y_H \subset H_0^1\left(\Omega\right)$), the coarse-scale problem is: find $p_H \in Y_H$ such that

$$\sum_{K \in \mathcal{T}_H} \int_K a\nabla p_H : \nabla q_H = \int_\Omega f q_H \quad \forall q_H \in Y_H \tag{2.22}$$

where q_H is the coarse-scale test functions.

Then we substitute $p_H(x)$ (2.21) into the above equation (2.22) and obtain a linear system of equations involving nodal values p_H^i. The system can be written in the matrix system:

$$\mathbf{A}_H \mathbf{P}_H = \mathbf{B}_H$$

where

$$\left(\mathbf{A}_H\right)_{ij} = \sum_{K \in \mathcal{T}_H} \int_K a\nabla\phi_i : \nabla\phi_j, \quad \left(\mathbf{B}_H\right)_i = \int_\Omega f\phi_i$$

with $\mathbf{P}_H = \left(p_H^i\right)$.

As in standard finite element method, the stiffness matrix $\mathbf{A}_H$ is a sparse matrix. Integrals in $\left(\mathbf{A}_H\right)_{ij}$ and $\left(\mathbf{B}_H\right)_i$ are computed locally on fine meshes, by projecting basis functions ϕ_i onto traditional finite element spaces. The basis functions and matrices are re-usable when the boundary condition changes. When the source term changes, we can re-use the pre-computed matrix $\mathbf{A}_H$ but we need to recompute the matrix $\mathbf{B}_H$.

2.5.4 Reconstruction of fine-scale solutions

After solving system (2.22), we obtain the coarse-scale solution. Then we reconstruct fine-scale features of the solution

$$p_H(x) = \sum_i p_H^i \phi_i(x) \quad \forall x \in \Omega$$

The fine-scale solutions are reconstructed locally on the fine mesh of each coarse element. Therefore one can choose to construct the fine-scale solution on the entire domain or only in the regions of interest. Moreover, the reconstruction can be done in parallel on different coarse elements, which allows to reduce computing time. In practice, this step is efficient due to the parallelism and the simplicity of the multiplication operation.

Nevertheless, the practical implementation of this step needs special consideration and may take important time. In fact, we reconstruct the fine-scale solutions by weighting the multiscale basis function ϕ_i with the corresponding coarse solution u_H^i. Multiscale basis functions are constructed on fine meshes whereas the coarse solution is obtained on the coarse mesh. For the basis function ϕ_i associated to node x_i, special attention needs to be paid to identify the corresponding coarse solution u_H^i. It is this correspondence between fine meshes and the coarse mesh which can take considerable time in practical implementation of this step.

2.6 The cost of multiscale finite element methods

If we discretize the domain Ω into M coarse elements and each coarse element into N sub-cell elements, there is a total of MN sub-cell elements. We denote $S(N)$ the flop counts (floating point operations) for solving a linear system with N degrees of freedom. Typically, we have $S(N) = \mathcal{O}(N^q)$ with $q > 1$.

Step	Cost
Discretization of the domain	$M \times \mathcal{O}(N)$
Solution of local problems	$M \times S(N)$
Computation of local matrices	$M \times \mathcal{O}(N)$
Solution of the coarse-scale problem	$S(M)$
Re-construction of fine-scale solutions	$M \times \mathcal{O}(N)$
Total cost	$M \times \mathcal{O}(N^q) + \mathcal{O}(M^q)$

TABLE 2.1: Estimation of computing memory

Solving the problem on a fine mesh with MN elements using a traditional finite element method needs about $\mathcal{S}(MN) = \mathcal{O}(M^q N^q)$ amount of computing memory. Table 2.1 shows that multiscale finite element methods require only about $M \times \mathcal{O}(M^q) + \mathcal{O}(N^q)$ amount of memory. Moreover, the computing time can be evaluated similarly to the computing memory. In multiscale finite element methods, the discretization of the domain, the solution of local problems, the computation of local matrices and the reconstruction of fine-scale solutions can be carried out on M coarse elements at the same time. Thus the total computing memory or computing time is reduced to $\mathcal{O}(M^q + N^q)$. Taking for example $M = N = 30$ and $q = 2$, then the traditional finite element requires 900 times more computing memory or time than multiscale finite element methods.

In multiscale finite element methods, basis functions are computed only once and are reusable in multiple solutions for different external parameters, such as boundary conditions or source terms. When using traditional finite element methods, however, each time an external parameter changes, we need to re-solve the problem on a very fine mesh which is very time-consuming. In conclusion, when solving the problem multiple times with different external parameters in the same medium, multiscale finite element methods allow significant computational savings compared to conventional numerical methods.

Chapter 3

Crouzeix-Raviart multiscale finite element methods

This chapter addresses several variants of Crouzeix-Raviart multiscale finite element methods to solve flow problems. In the literature, [95, 118] were the first to propose a Crouzeix-Raviart multiscale finite element method to solve Stokes problems and then [117] extended this method to Oseen problems. In these studies, the same flow problems were solved at both fine and coarse scales, i.e. multiscale basis functions are constructed as solutions of Stokes (or Oseen) equations on fine meshes and then used to solve Stokes (or Oseen) problems on the coarse mesh. Therefore, one question that needs to be raised is can multiscale basis functions be constructed as solutions of Stokes or Oseen problems at the fine scale and then used to solve Stokes, Oseen or Navier-Stokes problems on the coarse mesh. This question will be answered in this thesis. Moreover, another focus of this thesis is to propose various new methods to improve the accuracy of Crouzeix-Raviart multiscale finite element methods originally proposed in [95, 117, 118].

The chapter begins by reviewing the original Crouzeix-Raviart multiscale finite element method proposed in [95, 118] to solve Stokes problems. For the sake of simplicity, we denote this original method as CR-MsFEM-Stokes. The multiscale basis functions are constructed as solutions of Stokes equations at the fine scale. In this thesis, we propose to solve not only Stokes problems, but also Oseen problems and Navier-Stokes problems on the coarse mesh using this multiscale method. The Streamline-Upwind/Petrov-Galerkin (SUPG) stabilization technique [33] is adapted to the multiscale context in order to eliminate oscillations in the solution of coarse-scale problems.

In order to take inertial effects into account in multiscale basis functions, we present a Crouzeix-Raviart multiscale finite element method where local problems are defined by Oseen equations. This method was defined originally in [117] and will be denoted as CR-MsFEM-Oseen. Unfortunately, one drawback of the definition in [117] is that oscillations appear in the solution of local problems for large Oseen velocity. In order to solve this problem, we revise the definition of local problems suggested in [117] and prove the well-posedness of new local problems. We propose also to enrich this method by adding more weighting functions into the velocity approximation space. On the coarse mesh, we propose to solve Oseen problems as well as Navier-Stokes problems using this method. The SUPG stabilization technique is developed for the solution of coarse-scale problems.

The remaining part of this chapter seeks primarily to improve the accuracy of CR-MsFEM-Stokes and CR-MsFEM-Oseen. We propose several innovative methods to enrich the approximation spaces of velocity and pressure: (1) assembling velocity basis functions constructed in CR-MsFEM-Stokes and CR-MsFEM-Oseen together to form an enriched velocity space. (2) enriching the velocity space by adding bubble functions. The effects of bubble functions on the accuracy of the method are analyzed. (3) enriching both the velocity and pressure spaces by adding more weighting functions, leading to an innovative high-order Crouzeix-Raviart multiscale finite element method . The construction process of this method as well as the definition of local problems are presented in detail.

Outline Section 3.1 introduces the main idea of Crouzeix-Raviart multiscale finite element methods as well as basic definitions and notations. Section 3.2 and section 3.3 present Crouzeix-Raviart multiscale finite element methods where local problems are defined respectively by Stokes equations and Oseen equations. Section 3.4 presents the enrichment of the velocity space by multiscale basis functions previously defined in section 3.2 and section 3.3. Then section 3.5 discusses the enrichment of Crouzeix-Raviart multiscale finite element methods by bubble functions. Section 3.6 and section 3.7 present high-order Crouzeix-Raviart multiscale finite element methods where local problems are defined respectively by Stokes equations and Oseen equations.

3.1 Introduction

In this section, we introduce the main idea of Crouzeix-Raviart multiscale finite element methods and basic definitions and notations originally introduced in [95, 118]. The main steps of Crouzeix-Raviart multiscale finite element methods follow Algorithm 2.1. We focus on heterogeneous media with solid obstacles instead of oscillating permeability tensors as in section 2.5.

3.1.1 Discretization of the heterogeneous domain

Let Ω^ε be a perforated domain as shown in Figure 3.1 (a). The first step is to discretize the global domain Ω into a coarse mesh. We remind that Ω is an homogeneous domain which does not contain obstacles B^ε. We split the domain Ω into a regular mesh $\mathcal{T}_H$ with finite elements (triangles or quadrilaterals if $d = 2$, tetrahedrons or hexagons if $d = 3$) of size at most H. The size H is presumed to be much larger than the grid size necessary for a classical finite element method to solve flow problems in the domain Ω^ε. We note this partition as the coarse mesh and note finite elements of this partition as coarse elements.

Shown in Figure 3.1 (b), the coarse mesh $\mathcal{T}_H$ consists of N_H coarse elements K and N_E edges (if $d = 2$) or faces (if $d = 3$). Let $\mathcal{E}_H$ denote the set of edges or faces of $\mathcal{T}_H$, including those on the domain boundary $\partial\Omega$. For the sake of simplicity, all elements of $\mathcal{E}_H$ are referred as faces in both two ($d = 2$) and three ($d = 3$) dimensions. The coarse grid does not contain any hanging nodes and each face is shared by two neighboring elements except those on $\partial\Omega$ which belong to only one element.

For each coarse element K, we construct a finer mesh $\mathcal{T}_h(K)$ (see Figure 3.1 (c)) consisting of fine elements each with a width h. $\mathcal{T}_h(K)$ is noted as the fine mesh in this thesis. Typically $0 < h \ll H$ and $\mathcal{T}_h(K)$ is fine enough to fully resolve the boundary of obstacles B^ε. If fine meshes are matching along the interface between adjacent coarse elements, the union of fine meshes on all coarse elements forms a conforming fine mesh $\mathcal{T}_h(\Omega^\varepsilon)$ (see Figure 3.1 (d)), which overlaps with $\mathcal{T}_H$. We call $\mathcal{T}_h(\Omega^\varepsilon)$ the reference mesh in this thesis. We remind that $\mathcal{T}_h(\Omega^\varepsilon)$ is not required by multiscale finite element methods in practical applications. It is needed only to perform error analysis for the validation of multiscale methods.

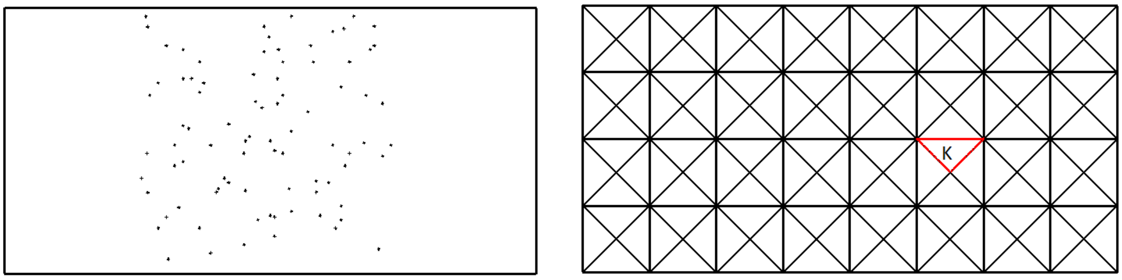
Let ω_E be the neighborhood of each face $E \in \mathcal{E}_H$ and it is defined by

$$\omega_E = \bigcup_j \{K_j \in \mathcal{T}_H \mid E \in \partial K_j\} \tag{3.1}$$

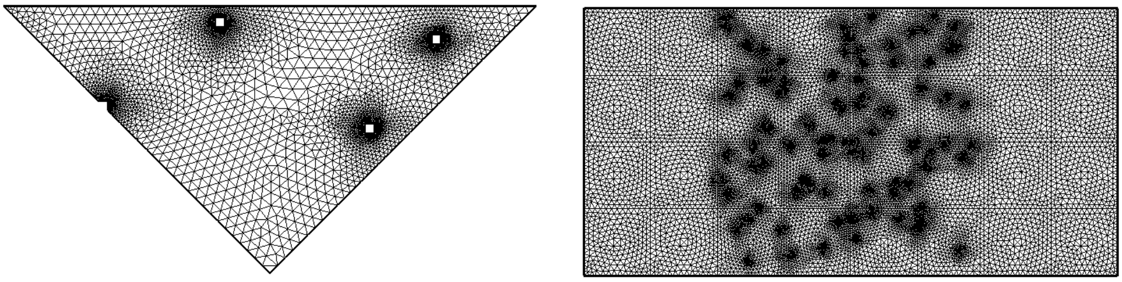
When E lies in the interior of the domain, ω_E is the union of two neighboring coarse elements.

3.1.2 Multiscale functional spaces

Let s be a positive integer and let $\boldsymbol{\omega}_{E,i}$: $E \to \mathbb{R}^d$ be some vector-valued functions associated to $E \in \mathcal{E}_H$ for $i = 1, \cdots, s$. In this thesis, we call these functions as weighting functions and they verify the assumption below.



(a) heterogeneous domain Ω^{ε} (b) coarse mesh $\mathcal{T}_H$ and element K



(c) coarse element $\mathcal{T}_h(K)$ (d) reference mesh $\mathcal{T}_h(\Omega^{\varepsilon})$

FIGURE 3.1: Illustration of the heterogeneous domain Ω^{ε}, the coarse mesh $\mathcal{T}_H$, one coarse element $\mathcal{T}_h(K)$ and the reference mesh $\mathcal{T}_h(\Omega^{\varepsilon})$

Assumption 3.1.1. *It is assumed that weighting functions satisfy*

$$\boldsymbol{n}_E \in \operatorname{span}\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\} \text{ for any } E \in \mathcal{E}_H$$

where $\boldsymbol{n}_E$ is the unit normal vector to E.

We recall important assumptions proposed in [95] which deal not only with the weights but also with the manner in which the holes B^{ε} intersect the mesh cells.

Assumption 3.1.2. *For any element $T \in \mathcal{T}_H$ and any real numbers $c_1^E, \cdots, c_s^E$ on all edges E composing ∂T. There exists $\boldsymbol{v} \in \left(H^1(T \cap \Omega^{\varepsilon})\right)^d$ vanishing on $\partial B^{\varepsilon} \cap T$ and such that*

$$\int_{E \cap \Omega^{\varepsilon}} \boldsymbol{v} \cdot \boldsymbol{\omega}_{E,i} = c_i^E$$

for all edges E and $i = 1, \cdots, s$.

Assumption 3.1.3. *For any $T \in \mathcal{T}_H$, let $C_1, \cdots, C_n$ be the connected components of $T \cap \Omega^{\varepsilon}$ and let any real numbers $c_1, \cdots, c_n$ with $\sum\limits_{i=1}^{n} c_i = 0$. There exists $\boldsymbol{w} \in \left(H^1\left(T \cap \Omega^{\varepsilon}\right)\right)^d$ vanishing on $\partial B^{\varepsilon} \cap T$ and such that $\int_{\partial C_i} \boldsymbol{w} \cdot \boldsymbol{n} = c_i$, $i = 1, \cdots, n$ and $\int_F \boldsymbol{w} \cdot \boldsymbol{\omega}_{F,j} = 0$ for all the edges F of T and $j = 1, \cdots, s$.*

[95] points out that Assumption 3.1.2 is usually valid provided that $\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}$ are linearly independent and the edge E is not covered completely by B^{ε}, which is the case of weighting functions in this thesis. In this thesis, we focus on the case where $T \cap \Omega^{\varepsilon}$ is connected for any $T \in \mathcal{T}_H$. Thus the assumption Assumption 3.1.3 is not considered in this work.

Definition 3.1.1. *We introduce the extended velocity space V_H^{ext}:*

$$V_H^{ext} = \left\{ \begin{array}{l} \boldsymbol{u} \in \left(L^2\left(\Omega^{\varepsilon}\right)\right)^d \text{ such that } \boldsymbol{u}|_T \in \left(H^1\left(T \cap \Omega^{\varepsilon}\right)\right)^d \text{ for any } T \in \mathcal{T}_H, \\ \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial B^{\varepsilon}, \ \int_{E \cap \Omega^{\varepsilon}} [[\boldsymbol{u}]] \cdot \boldsymbol{\omega}_{E,j} = 0 \text{ for all } E \in \mathcal{E}_H, \ j = 1, \cdots, s \end{array} \right\}$$

where $[[\boldsymbol{u}]]$ *denotes the "jump" of* $\boldsymbol{u}$ *across an internal face and* $[[\boldsymbol{u}]] = \boldsymbol{u}$ *on the boundary* $\partial\Omega$*.*

The space V_H^{ext} enhances the natural velocity space $\left(H_0^1\left(\Omega^\varepsilon\right)\right)^d$ so that we have at our disposal discontinuous vector fields across faces of the coarse mesh $\mathcal{T}_H$. The continuity of velocity across faces between adjacent coarse elements is maintained only in a weak sense, i.e., only the weighted average of $\boldsymbol{u}$ is continuous along the face between adjacent coarse elements. Therefore V_H^{ext} is not included in $\left(H_0^1\left(\Omega^\varepsilon\right)\right)^d$ and we construct a nonconforming multiscale finite element method. Thus we define a broken integral on V_H^{ext} in this thesis.

Definition 3.1.2. *Throughout this thesis, we use the broken integral defined on* V_H^{ext} *by:*

$$\int_{\Omega^\varepsilon} \nabla \boldsymbol{v} = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \nabla \boldsymbol{v},\ \forall \boldsymbol{v} \in V_H^{ext}$$

We recall that the pressure space M is defined by:

$$M = L_0^2\left(\Omega^\varepsilon\right) = \left\{p \in L^2\left(\Omega^\varepsilon\right) \text{ such that } \int_{\Omega^\varepsilon} p = 0\right\} \tag{3.2}$$

Then the extended velocity-pressure space X_H^{ext} can be written as

$$X_H^{ext} = V_H^{ext} \times M$$

As we are going to present in section 3.2, we want to decompose X_H^{ext} into a direct sum of a finite dimensional subspace X_H containing coarse scales and an infinite dimensional subspace X_H^0 containing unsolved fine scales.

$$X_H^{ext} = X_H \oplus X_H^0 \tag{3.3}$$

where $X_H^0 = V_H^0 \times M_H^0$ and M_H^0 is an infinite-dimensional pressure space. The subspace X_H will be used for approximation in numerical simulations.

Inspired by [95, 118], we propose several new methods to construct and to enrich the approximation space X_H, leading to different Crouzeix-Raviart multiscale finite element methods.

3.2 The Crouzeix-Raviart multiscale finite element method defined by Stokes equations

In this section, we first review the Crouzeix-Raviart multiscale finite element method originally proposed in [95, 118], where local problems are defined by Stokes equations. Then for the first time, we propose to solve not only Stokes problems (1.2), but also Oseen problems (1.3) and Navier-Stokes problems (1.1) by this method on the coarse mesh. We adapt the SUPG stabilization technique to the multiscale context in order to stabilize the solution. Moreover, the practical implementation of this multiscale method is very different to the original work. This is due to the fact that multiscale finite element methods are implemented in *TrioCFD* [139] which imposes some constraints to our work.

3.2.1 The construction of the approximation space X_H^{St}

We define the space of unresolved fine-scale features X_H^0 originally introduced in [95, 118]:

$$X_H^0 = V_H^0 \times M_H^0 \tag{3.4}$$

where

$$V_H^0 = \left\{ \boldsymbol{u} \in V_H^{ext} \text{ such that } \int_{E\cap\Omega^\varepsilon} \boldsymbol{u}\cdot\boldsymbol{\omega}_{E,j} = 0\ \forall E \in \mathcal{E}_H,\ \forall j = 1,\cdots,s \right\} \tag{3.5}$$

$$M_H^0 = \left\{ p \in M \text{ such that } \int_{T\cap\Omega^\varepsilon} p = 0\ \forall T \in \mathcal{T}_H \right\} \tag{3.6}$$

In two dimensions, [95, 118] propose weighting functions below: for any $E \in \mathcal{E}_H$,

$$\text{for } d = 2,\ \begin{cases} s = 2: \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2. \\ s = 3: \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2,\ \boldsymbol{\omega}_{E,3} = \boldsymbol{n}_E\psi_E. \end{cases} \tag{3.7}$$

where $\{\boldsymbol{e}_1,\ \boldsymbol{e}_2\}$ is the canonical basis of $\mathbb{R}^2$ and $\boldsymbol{n}_E$ is the unit vector normal to the face E. Besides, ψ_E is a linear polynomial which satisfies $\int_{E\cap\Omega^\varepsilon}\psi_E = 0$.

We propose in this thesis also the following weighting functions: for any $E \in \mathcal{E}_H$,

$$\text{for } d = 2,\ s = 4: \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2,\ \boldsymbol{\omega}_{E,3} = \boldsymbol{n}_E\psi_E,\ \boldsymbol{\omega}_{E,4} = \boldsymbol{\tau}_E\phi_E \tag{3.8}$$

$$\text{for } d = 3,\ \begin{cases} s = 3,\ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2,\ \boldsymbol{\omega}_{E,3} = \boldsymbol{e}_3 \\ s = 4: \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2,\ \boldsymbol{\omega}_{E,3} = \boldsymbol{e}_3,\ \boldsymbol{\omega}_{E,4} = \boldsymbol{n}_E\psi_E \end{cases} \tag{3.9}$$

where $\boldsymbol{n}_E$ and $\boldsymbol{\tau}_E$ are respectively the unit vector normal and tangent to the face E. Besides, ψ_E and ϕ_E are linear polynomials which satisfy $\int_{E\cap\Omega^\varepsilon}\psi_E = 0$ and $\int_{E\cap\Omega^\varepsilon}\phi_E = 0$.

Since this Crouzeix-Raviart multiscale finite element method is defined by Stokes equations, we denote the space X_H by X_H^{St} throughout this section.

Definition 3.2.1. *The space X_H^{St} is defined as the "orthogonal" complement of the space X_H^0 with respect to the bilinear form $c_H^{St}(.,.)$:*

$$(\boldsymbol{u}_H, p_H) \in X_H^{St} \Longleftrightarrow c_H^{St}\left((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)\right) = 0,\ \forall (\boldsymbol{v}, q) \in X_H^0 \tag{3.10}$$

where $c_H^{St}(.,.)$ is defined by:

$$c_H^{St}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} (\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p_H\, div\, \boldsymbol{v} - q\, div\, \boldsymbol{u}_H)$$

The word "orthogonal" is put between quotes as the bilinear form $c_H^{St}(.,.)$ is not a scalar product (not positive definite).

Definition 3.2.2. *Let the functional spaces $M_H \subset M$ and $V_H \subset V_H^{ext}$ be defined by:*

$$M_H = \left\{ q \in L_0^2(\Omega) \ \text{such that}\ q\,|_T = const,\ \forall T \in \mathcal{T}_H \right\} \tag{3.11}$$

$$V_H^{St} = \left\{ \begin{array}{l} \boldsymbol{v} \in \left(L^2(\Omega^\varepsilon)\right)^d: \ \forall T \in \mathcal{T}_H,\ \exists\zeta^T \in L_0^2(T\cap\Omega^\varepsilon),\ \exists\alpha^T \in \mathbb{R} \text{ such that} \\ -\mu\Delta\boldsymbol{v} + \nabla\zeta^T = \boldsymbol{0} \text{ in } T\cap\Omega^\varepsilon \\ div\, \boldsymbol{v} = \alpha^T \text{ in } T\cap\Omega^\varepsilon \\ \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T \\ \mu\nabla\boldsymbol{v}\boldsymbol{n} - \zeta^T\boldsymbol{n} \in \{\boldsymbol{\omega}_{E,1},\cdots,\boldsymbol{\omega}_{E,s}\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E \in \mathcal{E}(T) \end{array} \right\} \tag{3.12}$$

where $\mathcal{E}(T)$ is the set of faces composing ∂T and we recall

$$L_0^2(\Omega^\varepsilon) = \left\{ p \in L^2(\Omega^\varepsilon) \text{ such that } \int_{\Omega^\varepsilon} p = 0 \right\}$$
$$L_0^2(T \cap \Omega^\varepsilon) = \left\{ p \in L^2(T \cap \Omega^\varepsilon) \text{ such that } \int_{T\cap\Omega^\varepsilon} p = 0 \right\}$$

Definition 3.2.3. *For any $\boldsymbol{v} \in V_H^{St}$, on any element $T \in \mathcal{T}_H$, definition (3.12) implies that there exists a function ζ^T which is uniquely determined by $\boldsymbol{v}$. We can thus define a linear operator $\pi_H : V_H^{St} \to M_H^0$ such that for any $\boldsymbol{v} \in V_H^{St}$, $\pi_H(\boldsymbol{v}) = \zeta^T$ on any element $T \in \mathcal{T}_H$.*

Theorem 3.2.1. *Using M_H (3.11) and V_H^{St} (3.12), we define $\tilde{X}_H^{St}$ by*

$$\tilde{X}_H^{St} = \operatorname{span}\left\{ (\boldsymbol{u}_H,\ \pi_H(\boldsymbol{u}_H) + \bar{p}_H),\ \boldsymbol{u}_H \in V_H^{St},\ \bar{p}_H \in M_H \right\} \tag{3.13}$$

then the space X_H^{St} defined by (3.10) satisfies the following property

$$X_H^{St} = \tilde{X}_H^{St} \tag{3.14}$$

This theorem was first proposed and proved in [118]. Here we provide a more detailed proof in order to help the reader to understand the construction process of this and other Crouzeix-Raviart multiscale finite element methods presented later in this chapter.

Proof. We first prove that $(\boldsymbol{u}_H, p_H) \in X_H^{St}$ in the sense of definition (3.10) belongs to the space defined by (3.13). Let $(\boldsymbol{u}_H, p_H) \in X_H^{St}$ in the sense of definition (3.10), i.e. $\forall (\boldsymbol{v}, q) \in X_H^0$

$$c_H^{St}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} (\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} - p_H \operatorname{div} \boldsymbol{v} - q \operatorname{div} \boldsymbol{u}_H) = 0 \tag{3.15}$$

In a first step, for any element $T \in \mathcal{T}_H$, denoting $\bar{p}_H|_T$ the average of pressure p_H on this element. We define $\bar{p}_H$ a function whose restriction on T equals to $\bar{p}_H|_T$ on any element $T \in \mathcal{T}_H$. It is easy to see that $\bar{p}_H$ is well defined and $\bar{p}_H \in M_H$. Then $p'_H = (p_H - \bar{p}_H) \in M_H^0$ defined in (3.6). Consequently, we can decompose the pressure p_H in a unique way as

$$p_H = \bar{p}_H + p'_H \text{ with } \bar{p}_H \in M_H \text{ and } p'_H \in M_H^0$$

By virtue of this decomposition, the term concerning p_H in (3.15) can be decomposed as

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} p_H \operatorname{div} \boldsymbol{v} = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \bar{p}_H \operatorname{div} \boldsymbol{v} + \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} p'_H \operatorname{div} \boldsymbol{v} \tag{3.16}$$

Now we compute the first term in the right hand side of (3.16). As $\bar{p}_H$ is constant on each $T \in \mathcal{T}_H$, we have

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \bar{p}_H \operatorname{div} \boldsymbol{v} = \sum_{T\in\mathcal{T}_H} \bar{p}_H|_T \int_{T\cap\Omega^\varepsilon} \operatorname{div} \boldsymbol{v},\ \forall \boldsymbol{v} \in V_H^0$$

Besides, taking advantage of Assumption 3.1.1, for any $\boldsymbol{v} \in V_H^0$, the divergence theorem and the definition of V_H^0 reveals that

$$\int_{T\cap\Omega^\varepsilon} \operatorname{div} \boldsymbol{v} = \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v} \cdot \boldsymbol{n} = 0$$

As a result, the first term in the right hand side of (3.16) vanishes

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\bar{p}_H \text{ div } \boldsymbol{v}=0,\ \forall \boldsymbol{v}\in V_H^0$$

and then (3.15) reduces to: $\forall\left(\boldsymbol{v},q\right)\in X_H^0$

$$c_H^{St}\left(\left(\boldsymbol{u}_H,p_H\right),\left(\boldsymbol{v},q\right)\right)=\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v}-p_H' \text{ div } \boldsymbol{v}-q \text{ div } \boldsymbol{u}_H\right)=0 \tag{3.17}$$

In a second step, choosing an element $T\in\mathcal{T}_H$ and the test function $\boldsymbol{v}=\boldsymbol{0}$, for any $q\in M_H^0$ with q vanishing outside T, (3.17) becomes

$$\int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H=0$$

We deduce from this equation that there exists a constant α^T in $T\cap\Omega^\varepsilon$ such that

$$\text{div } \boldsymbol{u}_H=\alpha^T \text{ in } T\cap\Omega^\varepsilon \tag{3.18}$$

In a third step, we observe that for any face $E\in\mathcal{E}(T)$, there exist some (non unique) functions $\boldsymbol{v}_{E,i}\in\left(H^1\left(T\cap\Omega^\varepsilon\right)\right)^d,\ i=1,\cdots,s$ such that

$$\begin{cases}\int_{F\cap\Omega^\varepsilon}\boldsymbol{v}_{E,i}\cdot\boldsymbol{\omega}_{F,j}=\delta_{E,F}\delta_{i,j},\ \forall F\in\mathcal{E}(T),\ \forall j=1,\cdots,s\\ \boldsymbol{v}_{E,i}=\boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T\end{cases} \tag{3.19}$$

We denote by $V(T)$ the set of functions in $\left(H^1\left(T\cap\Omega^\varepsilon\right)\right)^d$ that vanish on $\partial B^\varepsilon\cap T$:

$$V(T)=\left\{\boldsymbol{v}\in\left(H^1(T\cap\Omega^\varepsilon)\right)^d \text{ such that } \boldsymbol{v}=\boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T\right\}$$

It is easy to check that $V(T)$ can be decomposed as

$$V(T)=V_{\int 0}(T)\oplus\text{span}\left\{\boldsymbol{v}_{E,i},\ \forall E\in\mathcal{E}(T),\ \forall i=1,\cdots,s\right\}$$

where

$$V_{\int 0}(T)=\left\{\begin{array}{l}\boldsymbol{v}\in\left(H^1(T\cap\Omega^\varepsilon)\right)^d:\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{E,i}=0,\ \forall E\in\mathcal{E}(T),\ \forall i=1,\cdots,s,\\ \boldsymbol{v}=\boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T\end{array}\right\}$$

By virtue of this decomposition, for any $\boldsymbol{v}\in V(T)$, there exist $\tilde{\boldsymbol{v}}\in V_{\int 0}(T)$ and $\beta_{F,1},\cdots,\beta_{F,s}\in\mathbb{R}$ such that

$$\boldsymbol{v}=\tilde{\boldsymbol{v}}+\sum_{F\in\mathcal{E}(T)}\sum_{j=1}^{s}\beta_{F,j}\boldsymbol{v}_{F,j} \tag{3.20}$$

Now let us compute the coefficient $\beta_{E,i}$ for $E\in\mathcal{E}(T)$ and $i=1,\cdots,s$. Multiplying (3.20) by $\boldsymbol{\omega}_{E,i}$ and integrating over E yields

$$\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{E,i}=\int_{E\cap\Omega^\varepsilon}\tilde{\boldsymbol{v}}\cdot\boldsymbol{\omega}_{E,i}+\sum_{F\in\mathcal{E}(T)}\sum_{j=1}^{s}\beta_{F,j}\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}_{F,j}\cdot\boldsymbol{\omega}_{E,i} \tag{3.21}$$

As $\tilde{\boldsymbol{v}} \in V_H^0$, definition of V_H^0 (3.5) implies that $\int_{E\cap\Omega^\varepsilon} \tilde{\boldsymbol{v}} \cdot \boldsymbol{\omega}_{E,i} = 0$. Besides, it is easy to deduce from (3.19) that

$$\sum_{F\in\mathcal{E}(T)} \sum_{j=1}^{s} \beta_{F,j} \int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_{F,j} \cdot \boldsymbol{\omega}_{E,i} = \beta_{E,i}$$

Finally (3.21) reduces to

$$\int_{E\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{E,i} = \beta_{E,i} \tag{3.22}$$

Let $\underline{\tilde{\boldsymbol{v}}}$ be the function defined on Ω^ε and equal to $\tilde{\boldsymbol{v}}$ on $T\cap\Omega^\varepsilon$ and $\mathbf{0}$ elsewhere. Hence it is obvious that $\underline{\tilde{\boldsymbol{v}}} \in V_H^0$. Taking $q = 0$, (3.17) shows that

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\underline{\tilde{\boldsymbol{v}}} - p'_H \text{ div } \underline{\tilde{\boldsymbol{v}}}\right) = \int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\tilde{\boldsymbol{v}} - p'_H \text{ div } \tilde{\boldsymbol{v}}\right) = 0$$

Substituting $\tilde{\boldsymbol{v}}$ defined in (3.20) into this equation, it is trivial to verify that

$$\int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p'_H \text{ div } \boldsymbol{v}\right) = \sum_{F\in\mathcal{E}(T)} \sum_{j=1}^{s} \beta_{F,j} \int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{F,j} - p'_H \text{ div } \boldsymbol{v}_{F,j}\right) \tag{3.23}$$

Denoting for any $F \in \mathcal{E}(T)$ and $j = 1, \cdots, s$

$$\lambda_{F,j} = \int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{F,j} - p'_H \text{ div } \boldsymbol{v}_{F,j}\right)$$

and substituting $\beta_{F,j}$ (3.22) into (3.23), we obtain

$$\int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p'_H \text{ div } \boldsymbol{v}\right) = \sum_{F\in\mathcal{E}(T)} \sum_{j=1}^{s} \lambda_{F,j} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{F,j},\ \forall\boldsymbol{v} \in V(T) \tag{3.24}$$

In (3.24), taking $\boldsymbol{v} = 0$ on $\mathcal{E}(T)$ and integrating by parts the left hand side, we obtain

$$-\mu\Delta\boldsymbol{u}_H + \nabla p'_H = \mathbf{0} \text{ in } T\cap\Omega^\varepsilon \tag{3.25}$$

Then by writing the variational formulation of (3.25) for $\boldsymbol{v} \in V(T)$ and comparing with (3.24), we deduce that

$$\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - p'_H\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E \in \mathcal{E}(T) \tag{3.26}$$

Finally, combining equations (3.18), (3.25) and (3.26), we obtain the following system

$$\begin{aligned}
-\mu\Delta\boldsymbol{u}_H + \nabla p'_H &= \mathbf{0} \text{ in } T\cap\Omega^\varepsilon \\
\text{div } \boldsymbol{u}_H &= \alpha^T \text{ in } T\cap\Omega^\varepsilon \\
\boldsymbol{u}_H &= \mathbf{0} \text{ on } \partial B^\varepsilon \cap T \\
\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - p'_H\boldsymbol{n} &\in \text{span}\left\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E \in \mathcal{E}(T)
\end{aligned}$$

On any element $T \in \mathcal{T}_H$, for any $\boldsymbol{u}_H \in V_H^{St}$ fixed in the formula above, it is easy to see that the gradient $\nabla p'_H$ is uniquely determined by the first equation. Besides, the fact that the average of p'_H is 0 over any element implies that p'_H is uniquely determined by $\boldsymbol{u}_H$. Thus we conclude that there exists a linear operator π_H such that $p'_H = \pi_H(\boldsymbol{u}_H)$ with $\boldsymbol{u}_H \in V_H^{St}$. We recall that the pressure p_H is decomposed as $p_H = \bar{p}_H + p'_H$. Thus we have proved that the solution $(\boldsymbol{u}_H, p_H)$ defined by (3.10) belongs to the space defined by (3.13).

Reciprocally, we now prove that any $(\boldsymbol{u}_H, p_H) \in \tilde{X}_H^{St}$ defined by (3.13) satisfies also the relation (3.10). Let $(\boldsymbol{u}_H, p_H) \in \tilde{X}_H^{St}$ defined by (3.13). For any $\boldsymbol{u}_H \in V_H^{St}$, on any $T \in \mathcal{T}_H$, definition of V_H^{St} (3.12) reveals that there exist a unique $\zeta^T = \pi_H(\boldsymbol{u}_H) \in L_0^2(T \cap \Omega^\varepsilon)$ and a unique constant $\alpha^T \in \mathbb{R}$ such that

$$
\begin{aligned}
-\mu \Delta \boldsymbol{u}_H + \nabla \zeta^T &= \mathbf{0} \text{ in } T \cap \Omega^\varepsilon & (3.27)\\
\text{div } \boldsymbol{u}_H &= \alpha^T \text{ in } T \cap \Omega^\varepsilon & (3.28)\\
\boldsymbol{u}_H &= \mathbf{0} \text{ on } \partial B^\varepsilon \cap T &\\
\mu \nabla \boldsymbol{u}_H \boldsymbol{n} - \zeta^T \boldsymbol{n} &\in \text{span}\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\} \text{ on } E \cap \Omega^\varepsilon,\ \forall E \in \mathcal{E}(T) & (3.29)
\end{aligned}
$$

For any $\boldsymbol{v} \in V_H^0$, integrating by parts (3.27) yields:

$$\int_{T\cap\Omega^\varepsilon} \mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = \int_{\partial(T\cap\Omega^\varepsilon)} \left(\mu \nabla \boldsymbol{u}_H \boldsymbol{n} - \zeta^T \boldsymbol{n}\right) \cdot \boldsymbol{v} \tag{3.30}$$

Equation (3.29) shows that for any $E \in \mathcal{E}(T)$, there exist $\lambda_{E,1}, \cdots, \lambda_{E,s} \in \mathbb{R}$ such that

$$\mu \nabla \boldsymbol{u}_H \boldsymbol{n} - \zeta^T \boldsymbol{n} = \sum_{i=1}^{s} \lambda_{E,i} \boldsymbol{\omega}_{E,i} \text{ on } E \cap \Omega^\varepsilon$$

Substituting this expression into (3.30), we obtain

$$\int_{T\cap\Omega^\varepsilon} \mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = \sum_{E\in\mathcal{E}(T)} \sum_{i=1}^{s} \lambda_{E,i} \int_{E\cap\Omega^\varepsilon} \boldsymbol{\omega}_{E,i} \cdot \boldsymbol{v} \tag{3.31}$$

Moreover, for any $\boldsymbol{v} \in V_H^0$, definition of V_H^0 (3.5) implies that

$$\sum_{E\in\mathcal{E}(T)} \sum_{i=1}^{s} \lambda_{E,i} \int_{E\cap\Omega^\varepsilon} \boldsymbol{\omega}_{E,i} \cdot \boldsymbol{v} = 0$$

Consequently, (3.31) reduces to

$$\int_{T\cap\Omega^\varepsilon} \mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = 0 \tag{3.32}$$

Besides, taking advantage of Assumption 3.1.1, for any $\boldsymbol{v} \in V_H^0$, the divergence theorem and definition of V_H^0 (3.5) imply that

$$\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = \bar{p}_H|_T \int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{v} = \bar{p}_H|_T \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v} \cdot \boldsymbol{n} = 0 \tag{3.33}$$

Meanwhile, for any $q \in M_H^0$, (3.28) shows that

$$\int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0 \tag{3.34}$$

As a result, summing (3.32), (3.33) and (3.34), we obtain: $\forall\,(\boldsymbol{v},q)\in V_H^0\times M_H^0$

$$\int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v}-\int_{T\cap\Omega^\varepsilon}\left(\zeta^T+\bar{p}_H|_T\right)\,\mathrm{div}\,\boldsymbol{v}-\int_{T\cap\Omega^\varepsilon}q\,\mathrm{div}\,\boldsymbol{u}_H=0$$

Denoting $p_H|_T=\zeta^T+\bar{p}_H|_T$ on each element $T\in\mathcal{T}_H$, let p_H be a function which equals to $p_H|_T$ on each $T\in\mathcal{T}_H$, then $(\boldsymbol{u}_H,p_H)\in X_H^{St}$. Summing this equation over all elements $T\in\mathcal{T}_H$, we obtain that $\forall\,(\boldsymbol{v},q)\in X_H^0$

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v}-\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}p_H\,\mathrm{div}\,\boldsymbol{v}-\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}q\,\mathrm{div}\,\boldsymbol{u}_H=0$$

which is exactly definition (3.10).

Finally, we have proved the identity between (3.13) and (3.10), i.e. property (3.14). ☐

Theorem 3.2.2. *The space X_H^{ext} can be decomposed using X_H^0 (3.4) and X_H^{St} (3.10) as:*

$$X_H^{ext}=X_H^{St}\oplus X_H^0$$

Proof. This theorem can be proved in two steps:

$$(i)\;X_H^{ext}=X_H^{St}+X_H^0,\quad(ii)\;X_H^{St}\cap X_H^0=\{0\}$$

Now we start by proving the step (i), i.e. for any $(\boldsymbol{u},p)\in X_H^{ext}$, there exist $\boldsymbol{u}_H\in V_H^{St}$, $\bar{p}_H\in M_H$, $\boldsymbol{u}^0\in V_H^0$ and $p^0\in M_H^0$ such that

$$\boldsymbol{u}=\boldsymbol{u}_H+\boldsymbol{u}^0,\;p=\pi_H\left(\boldsymbol{u}_H\right)+\bar{p}_H+p^0\tag{3.35}$$

To prove the step (i), we first prove that for any $(\boldsymbol{u},p)\in X_H^{ext}$, there exist $\boldsymbol{u}^0\in V_H^0$ and $p^0\in M_H^0$ such that

$$c_H^{St}\left(\left(\boldsymbol{u}^0,p^0\right),(\boldsymbol{v},q)\right)=c_H^{St}\left((\boldsymbol{u},p),(\boldsymbol{v},q)\right),\;\forall\,(\boldsymbol{v},q)\in V_H^0\times M_H^0$$

In order to prove the existence of such $\left(\boldsymbol{u}^0,p^0\right)\in V_H^0\times M_H^0$, we pick up any triangle $T\in\mathcal{T}_H$ and notice that the restriction of $\left(\boldsymbol{u}^0,p^0\right)$ to the triangle T belongs to $V_{\int 0}(T)\times L_0^2\left(T\cap\Omega^\varepsilon\right)$. By restricting test functions $\boldsymbol{v}$ to $V_{\int 0}(T)$ and q to $L_0^2\left(T\cap\Omega^\varepsilon\right)$, it is easy to check that $\left(\boldsymbol{u}^0,p^0\right)$ should satisfy

$$\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}^0:\nabla\boldsymbol{v}-p^0\,\mathrm{div}\,\boldsymbol{v}\right)=\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}:\nabla\boldsymbol{v}-p\,\mathrm{div}\,\boldsymbol{v}\right),\;\forall\boldsymbol{v}\in V_{\int 0}(T)\tag{3.36}$$

$$\int_{T\cap\Omega^\varepsilon}q\,\mathrm{div}\,\boldsymbol{u}^0=\int_{T\cap\Omega^\varepsilon}q\,\mathrm{div}\,\boldsymbol{u},\;\forall q\in L_0^2\left(T\cap\Omega^\varepsilon\right)\tag{3.37}$$

This is a standard saddle point problem and the existence of its solution $\left(\boldsymbol{u}^0,p^0\right)\in V_{\int 0}(T)\times L_0^2\left(T\cap\Omega^\varepsilon\right)$ is guaranteed by the inf-sup property proved in [118]:

$$\inf_{q\in L_0^2(T\cap\Omega^\varepsilon)}\sup_{\boldsymbol{v}\in V_{\int 0}(T)}\frac{\int_{T\cap\Omega^\varepsilon}q\,\mathrm{div}\,\boldsymbol{v}}{\|q\|_{L^2(T\cap\Omega^\varepsilon)}\|\boldsymbol{v}\|_{H^1(T)}}>0$$

Therefore we have proved the existence of $\left(\boldsymbol{u}^0,p^0\right)\in V_{\int 0}(T)\times L_0^2\left(T\cap\Omega^\varepsilon\right)$. By gluing together $\left(\boldsymbol{u}^0,p^0\right)$ on each element $T\in\mathcal{T}_H$, we obtain that $\left(\boldsymbol{u}^0,p^0\right)\in V_H^0\times M_H^0$. Consequently, we have proved that for any $(\boldsymbol{u},p)\in X_H^{ext}$, there exist $\left(\boldsymbol{u}^0,p^0\right)\in X_H^0$.

Then we define $\boldsymbol{u}_H$ and p_H by

$$\boldsymbol{u}_H = \boldsymbol{u} - \boldsymbol{u}^0, \quad p_H = p - p^0$$

It is trivial to check that $(\boldsymbol{u}_H, p_H)$ satisfies

$$c_H^{St}\left(\left(\boldsymbol{u}_H, p_H\right), (\boldsymbol{v}, q)\right) = 0, \ \forall (\boldsymbol{v}, q) \in V_H^0 \times M_H^0$$

Making use of Theorem 3.2.1, we deduce that $(\boldsymbol{u}_H, p_H) \in V_H^{St} \times M_H$. Finally, we conclude that $X_H^{ext} = X_H^{St} + X_H^0$.

Now it remains to prove that $X_H^{St} \cap X_H^0 = \{0\}$. Let $(\boldsymbol{u}, p) \in X_H^{ext}$ and $(\boldsymbol{u}, p) \in X_H^{St} \cap X_H^0$, then Theorem 3.2.1 implies that

$$c_H^{St}\left(\left(\boldsymbol{u}, p\right), (\boldsymbol{v}, q)\right) = 0 \tag{3.38}$$

for $(\boldsymbol{v}, q) \in X_H^{St}$ as well as for $(\boldsymbol{v}, q) \in X_H^0$. Meanwhile, the fact that $X_H^{ext} = X_H^{St} + X_H^0$ implies that (3.38) holds true for any $(\boldsymbol{v}, q) \in X_H^{ext}$. Consequently, we can deduce that $(\boldsymbol{u}, p) = 0$ using the inf-sup property of Theorem 2.1.2.

Therefore, combining the results above, we conclude that $X_H^{ext} = X_H \oplus X_H^0$. ☐

3.2.2 The local problem defined by Stokes equations

We now present local problems defined by Stokes equations which were originally proposed in [95, 118].

The strong form For any face $E \in \mathcal{E}_H$, $i = 1, \cdots, s$, we construct $\boldsymbol{\Phi}_{E,i} : \Omega^\varepsilon \to \mathbb{R}^d$ and the accompanying pressure $\pi_{E,i} : \Omega^\varepsilon \to \mathbb{R}$ such that $\boldsymbol{\Phi}_{E,i}$ and $\pi_{E,i}$ vanish outside the two coarse elements $T_k \subset \omega_E$, $k = \{1, 2\}$ (only one coarse element if $E \in \partial\Omega$) and solve on T_k:

$$\begin{cases} -\mu \Delta \boldsymbol{\Phi}_{E,i} + \nabla \pi_{E,i} = \mathbf{0} \text{ in } T_k \cap \Omega^\varepsilon, \\ \operatorname{div} \boldsymbol{\Phi}_{E,i} = \alpha_{E,i} \text{ in } T_k \cap \Omega^\varepsilon, \\ \mu \nabla \boldsymbol{\Phi}_{E,i} \boldsymbol{n} - \pi_{E,i} \boldsymbol{n} \in \operatorname{span}\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\} \text{ on } F \cap \Omega^\varepsilon, \ \forall F \in \mathcal{E}(T_k), \\ \boldsymbol{\Phi}_{E,i} = \mathbf{0} \text{ on } \partial B^\varepsilon \cap T_k, \\ \int_{F \cap \Omega^\varepsilon} \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{\omega}_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases} \quad \forall F \in \mathcal{E}\left(T_k\right), \ j = 1, \cdots, s. \\ \int_{T_k \cap \Omega^\varepsilon} \pi_{E,i} = 0. \end{cases} \tag{3.39}$$

where $\mathcal{E}(T_k)$ is the set of faces of T_k. The constant $\alpha_{E,i}$ depends on T_k and satisfies the compatibility relation $\int_{T_k \cap \Omega^\varepsilon} \alpha_{E,i} = \int_{\partial(T_k \cap \Omega^\varepsilon)} \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{n}$.

To be more explicit, we illustrate the local problem (3.39) in Figure 3.2. We can see that for an internal face $E \in \mathcal{E}_H$, its support is $\omega_E = \{T_1, T_2\}$. In both T_1 and T_2, we solve Stokes equations with integral type boundary conditions. We notice that the divergence of velocity is different in T_1 and T_2.

The weak form The weak form of system (3.39) reads: for any face $E \in \mathcal{E}_H$, $i = 1, \cdots, s$, on the coarse element $T_k \subset \omega_E$ for $k = \{1, 2\}$ (only one coarse element if $E \in \partial\Omega$), find $\boldsymbol{\Phi}_{E,i} \in \left(H^1\left(T_k \cap \Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{\Phi}_{E,i} = \mathbf{0}$ on $\partial B^\varepsilon \cap T_k$, $\pi_{E,i} \in L_0^2\left(T_k \cap \Omega^\varepsilon\right)$ and Lagrange

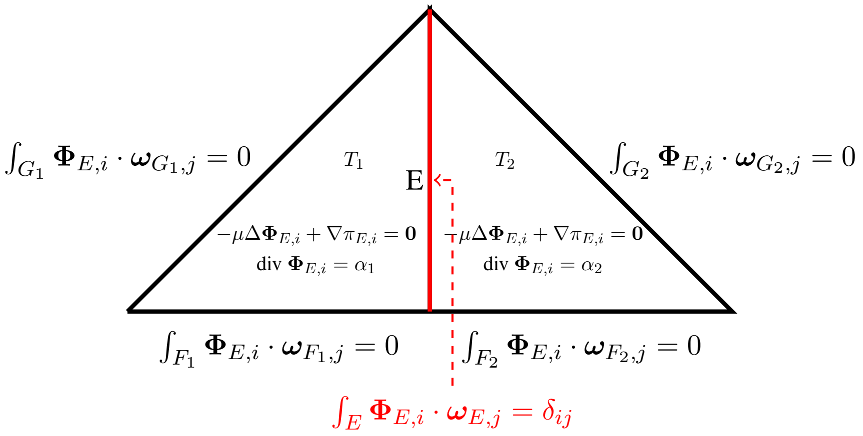


FIGURE 3.2: Schematic illustration of basis functions associated to a face E between T_1 and T_2

multipliers $\lambda_{F,1}, \cdots, \lambda_{F,s} \in \mathbb{R}$ for $F \in \mathcal{E}(T_k)$ such that

$$\begin{aligned} \int_{T_k\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{\Phi}_{E,i} : \nabla\boldsymbol{v} - \int_{T_k\cap\Omega^\varepsilon} \pi_{E,i} \operatorname{div} \boldsymbol{v} + \sum_{F\in\mathcal{E}(T_k)} \sum_{j=1}^{s} \lambda_{F,j} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v}\cdot\boldsymbol{\omega}_{F,j} &= 0, \\ \int_{T_k\cap\Omega^\varepsilon} q \operatorname{div} \boldsymbol{\Phi}_{E,i} &= 0, \\ \sum_{F\in\mathcal{E}(T_k)} \sum_{j=1}^{s} \mu_{F,j} \int_{F\cap\Omega^\varepsilon} \boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\omega}_{F,j} &= \mu_{E,i} \end{aligned}$$

for all $\boldsymbol{v} \in \left(H^1\left(T_k\cap\Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{v} = \mathbf{0}$ on $\partial B^\varepsilon \cap T_k$, $q \in L_0^2\left(T_k\cap\Omega^\varepsilon\right)$, $\mu_{F,j} \in \mathbb{R}$ for all $F \in \mathcal{E}\left(T_k\right)$ and $j = 1, \cdots, s$.

Remark. *When weighting functions are chosen as the case $s = d$ defined in (3.7) and (3.9), the space V_H^{St} is reduced to the standard Crouzeix-Raviart finite element space (see subsection 2.4.1) if there are no obstacles in the domain, i.e. $B^\varepsilon = \varnothing$. In this situation, it is obvious that functions constructed in (3.39) can be written as $\boldsymbol{\Phi}_{E,i} = \Phi_E\boldsymbol{e}_i$ for $E \in \mathcal{E}_H$ and $i = 1, \cdots, d$ where Φ_E is linear on any element $T \in \mathcal{T}_H$, discontinuous across the face E and satisfies that $\int_E \Phi_E = 1$ and $\int_F \Phi_E = 0$ for all $F \in \mathcal{E}_H$, $F \neq E$. Besides, the pressure $\pi_{E,i} = 0$ for $i = 1, \cdots, d$.*

3.2.3 The basis function of the space V_H^{St}

Theorem 3.2.3. *The functions $\boldsymbol{\Phi}_{E,i}$ for $E \in \mathcal{E}_H$ and $i = 1, \cdots, s$ constructed in (3.39) form a basis of V_H^{St} defined by (3.12). In other words,*

$$V_H^{St} = \operatorname{span}\{\boldsymbol{\Phi}_{E,i},\ E \in \mathcal{E}_H,\ i = 1, \cdots, s\}. \tag{3.40}$$

This theorem was first proposed in [118] but no detailed proof was provided. Now we provide a proof of this theorem in what follows.

Proof. Firstly, it is easy to check that $\operatorname{span}\{\boldsymbol{\Phi}_{E,i},\ E \in \mathcal{E}_H,\ i = 1, \cdots, s\} \subset V_H^{St}$. It remains to prove that $V_H^{St} \subset \operatorname{span}\{\boldsymbol{\Phi}_{E,i},\ E \in \mathcal{E}_H,\ i = 1, \cdots, s\}$. Let $\boldsymbol{u} \in V_H^{St}$, then definition of V_H^{St} (3.12) implies that: for any $T \in \mathcal{T}_H$, there exist a unique $\zeta^T \in L_0^2\left(T\cap\Omega^\varepsilon\right)$ and a unique

$\alpha^T \in \mathbb{R}$ such that

$$\begin{aligned} -\mu\Delta \boldsymbol{u} + \nabla\zeta^T &= \mathbf{0} \text{ in } T\cap\Omega^\varepsilon \\ \text{div } \boldsymbol{u} &= \alpha^T \text{ in } T\cap\Omega^\varepsilon \\ \boldsymbol{u} &= \mathbf{0} \text{ on } \partial B^\varepsilon \cap T \\ \mu\nabla\boldsymbol{u}\boldsymbol{n} - \zeta^T\boldsymbol{n} &\in \text{span}\,\{\boldsymbol{\omega}_{E,1},\cdots,\boldsymbol{\omega}_{E,s}\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E\in\mathcal{E}(T) \end{aligned}$$

For any $T \in \mathcal{T}_H$, we introduce $\boldsymbol{v}$ and σ defined by

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{u} - \sum_{E\in\mathcal{E}(T)}\sum_{i=1}^{s}\left(\int_{E\cap\Omega^\varepsilon}\boldsymbol{u}\cdot\boldsymbol{\omega}_{E,i}\right)\boldsymbol{\Phi}_{E,i} \\ \sigma &= \zeta^T - \sum_{E\in\mathcal{E}(T)}\sum_{i=1}^{s}\left(\int_{E\cap\Omega^\varepsilon}\boldsymbol{u}\cdot\boldsymbol{\omega}_{E,i}\right)\pi_{E,i} \end{aligned}$$

It is easy to check that there exists a constant $\beta^T \in \mathbb{R}$ such that $\boldsymbol{v}$ and σ satisfy

$$-\mu\Delta\boldsymbol{v} + \nabla\sigma = \mathbf{0} \text{ on } T\cap\Omega^\varepsilon \tag{3.41}$$

$$\text{div } \boldsymbol{v} = \beta^T \text{ on } T\cap\Omega^\varepsilon \tag{3.42}$$

$$\boldsymbol{v} = \mathbf{0} \text{ on } \partial B^\varepsilon\cap T$$

$$\mu\nabla\boldsymbol{v}\boldsymbol{n} - \sigma\boldsymbol{n} \in \text{span}\,\{\boldsymbol{\omega}_{E,1},\cdots,\boldsymbol{\omega}_{E,s}\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E\in\mathcal{E}(T) \tag{3.43}$$

$$\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{E,i} = 0\ \forall E\in\mathcal{E}(T),\ \forall i = 1,\cdots,s \tag{3.44}$$

$$\int_{T\cap\Omega^\varepsilon}\sigma = 0 \tag{3.45}$$

where the constant β^T satisfies $\int_{T\cap\Omega^\varepsilon}\beta^T = \int_{\partial(T\cap\Omega^\varepsilon)}\boldsymbol{v}\cdot\boldsymbol{n}$.

Taking $\boldsymbol{v}$ as the test function, the variational formulation of (3.41) is

$$\int_{T\cap\Omega^\varepsilon}\mu|\nabla\boldsymbol{v}|^2 - \int_{T\cap\Omega^\varepsilon}\sigma\text{ div }\boldsymbol{v} = \int_{\partial(T\cap\Omega^\varepsilon)}(\mu\nabla\boldsymbol{v}\boldsymbol{n} - \sigma\boldsymbol{n})\cdot\boldsymbol{v} \tag{3.46}$$

Then (3.42) and (3.45) imply that

$$\int_{T\cap\Omega^\varepsilon}\sigma\text{ div }\boldsymbol{v} = 0$$

Besides, (3.43)–(3.44) show that

$$\int_{\partial(T\cap\Omega^\varepsilon)}(\mu\nabla\boldsymbol{v}\boldsymbol{n} - \sigma\boldsymbol{n})\cdot\boldsymbol{v} = 0$$

Finally (3.46) reduces to

$$\int_{T\cap\Omega^\varepsilon}\mu|\nabla\boldsymbol{v}|^2 = 0$$

which reveals that $\boldsymbol{v}$ is constant on $T\cap\Omega^\varepsilon$. On combining (3.44), we deduce that $\boldsymbol{v} = \mathbf{0}$ and then $\sigma = 0$.

The fact that $\boldsymbol{v} = \mathbf{0}$ and $\sigma = 0$ imply that for any element $T \in \mathcal{T}_H$, $\boldsymbol{u}$ and ζ^T can be represented as

$$\boldsymbol{u} = \sum_{E \in \mathcal{E}(T)} \sum_{i=1}^{s} \left(\int_{E \cap \Omega^\varepsilon} \boldsymbol{u} \cdot \boldsymbol{\omega}_{E,i} \right) \boldsymbol{\Phi}_{E,i}$$

$$\zeta^T = \sum_{E \in \mathcal{E}(T)} \sum_{i=1}^{s} \left(\int_{E \cap \Omega^\varepsilon} \boldsymbol{u} \cdot \boldsymbol{\omega}_{E,i} \right) \pi_{E,i}$$

which implies that $V_H^{St} \subset \operatorname{span}\{\boldsymbol{\Phi}_{E,i},\ E \in \mathcal{E}_H,\ i = 1, \cdots, s\}$. Combining the results above, we conclude that $V_H^{St} = \operatorname{span}\{\boldsymbol{\Phi}_{E,i},\ E \in \mathcal{E}_H,\ i = 1, \cdots, s\}$. ☐

Consequently, any function of V_H^{St} can be written with multiscale basis functions as

$$\forall \boldsymbol{u}_H \in V_H^{St},\ \boldsymbol{u}_H = \sum_{E \in \mathcal{E}_H} \sum_{i=1}^{s} u_{E,i} \boldsymbol{\Phi}_{E,i}$$

We obtain also an explicit formulation of the linear operator π_H (see Definition 3.2.3):

$$\forall \boldsymbol{u}_H \in V_H^{St},\ \pi_H \left(\boldsymbol{u}_H \right) = \sum_{E \in \mathcal{E}_H} \sum_{i=1}^{s} u_{E,i} \pi_{E,i} \tag{3.47}$$

3.2.4 The coarse-scale problems and stabilized formulations

In [95, 118], only Stokes problems were solved on the coarse mesh using the approximation space X_H^{St}. In this thesis, we propose to solve not only Stokes problems, but also Oseen problems and Navier-Stokes problems on the coarse mesh using X_H^{St}.

3.2.4.1 The discrete variational formulation of the Stokes problem

The discrete variational formulation of Stokes problem (1.2) is: find $(\boldsymbol{u}_H, p_H) \in X_H^{St}$ such that

$$c_H^{St} \left((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q) \right) = (\boldsymbol{f}, \boldsymbol{v}),\ \forall (\boldsymbol{v}, q) \in X_H^{St} \tag{3.48}$$

Theorem 3.2.1 reveals that p_H can be decomposed as $p_H = \pi_H (\boldsymbol{u}_H) + \bar{p}_H$ with $\bar{p}_H \in M_H$. Besides, it is trivial to verify that $(\pi_H (\boldsymbol{u}_H), \operatorname{div} \boldsymbol{v}) = 0$ for all $\boldsymbol{u}_H, \boldsymbol{v} \in V_H^{St}$. Using this property, (3.48) can be reformulated as: find $\boldsymbol{u}_H \in V_H^{St}$ and $\bar{p}_H \in M_H$ such that

$$a_H^{St} (\boldsymbol{u}_H, \boldsymbol{v}) + b_H (\boldsymbol{v}, \bar{p}_H) = F_H (\boldsymbol{v}),\ \forall \boldsymbol{v} \in V_H^{St} \tag{3.49}$$

$$b_H (\boldsymbol{u}_H, q) = 0,\ \forall q \in M_H \tag{3.50}$$

where

$$a_H^{St} (\boldsymbol{u}_H, \boldsymbol{v}) = \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} \tag{3.51}$$

$$b_H (\boldsymbol{v}, \bar{p}_H) = - \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \bar{p}_H \operatorname{div} \boldsymbol{v} \tag{3.52}$$

$$F_H (\boldsymbol{v}) = \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \boldsymbol{f} \cdot \boldsymbol{v} \tag{3.53}$$

We now discuss the well-posedness of discrete variational problem (3.49)–(3.50). We first propose the following theorem.

Theorem 3.2.4. *The space V_H^{St} and M_H have the following relation:*

$$\mathit{div}\ \left(V_H^{St}\right) = M_H$$

Proof. We first prove that $\operatorname{div}\left(V_H^{St}\right) \subset M_H$. For any $\boldsymbol{v}_H \in V_H^{St}$, $\operatorname{div}(\boldsymbol{v}_H)$ is a piecewise constant on $\mathcal{T}_H$ and it is easy to verify that

$$\int_{\Omega^\varepsilon} \operatorname{div} \boldsymbol{v}_H = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \operatorname{div} \boldsymbol{v}_H = \sum_{E\in\mathcal{E}_H} \int_{E\cap\Omega^\varepsilon} [[\boldsymbol{v}_H \cdot \boldsymbol{n}_E]] = 0$$

Therefore we have proved that $\operatorname{div}\left(V_H^{St}\right) \subset M_H$.

Reciprocally, for any $q_H \in M_H$, [134] implies that there exists $\boldsymbol{v} \in \left(H_0^1\left(\Omega^\varepsilon\right)\right)^d$ such that $\operatorname{div} \boldsymbol{v} = q$. We observe that $\boldsymbol{v}$ can be decomposed as

$$\boldsymbol{v} = \boldsymbol{v}_H + \boldsymbol{v}_H^0 \text{ with } \boldsymbol{v}_H \in V_H^{St} \text{ and } \boldsymbol{v}_H^0 \in V_H^0$$

Multiplying this equation by $\boldsymbol{n}_E$ and integrating on a face $E \in \mathcal{E}_H$, we obtain

$$\int_{E\cap\Omega^\varepsilon} \boldsymbol{v}\cdot\boldsymbol{n}_E = \int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_H\cdot\boldsymbol{n}_E + \int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_H^0\cdot\boldsymbol{n}_E \tag{3.54}$$

Making use of Assumption 3.1.1, the definition of V_H^0 implies that $\int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_H^0 \cdot \boldsymbol{n}_E = 0$. Hence (3.54) reduces to

$$\int_{E\cap\Omega^\varepsilon} \boldsymbol{v}\cdot\boldsymbol{n}_E = \int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_H\cdot\boldsymbol{n}_E,\ \forall E\in\mathcal{E}_H$$

Using this relation, we deduce that: for any $T \in \mathcal{T}_H$,

$$\int_{T\cap\Omega^\varepsilon} \operatorname{div} \boldsymbol{v}_H = \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v}_H\cdot\boldsymbol{n} = \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v}\cdot\boldsymbol{n} = \int_{T\cap\Omega^\varepsilon} \operatorname{div} \boldsymbol{v} = \int_{T\cap\Omega^\varepsilon} q$$

Since both $\operatorname{div} \boldsymbol{v}_H$ and q are piecewise constant on $\mathcal{T}_H$, we conclude that $\operatorname{div} \boldsymbol{v}_H = q$ and thus $M_H \subset \operatorname{div}\left(V_H^{St}\right)$.

In conclusion, combining the results above, we have proved that $\operatorname{div}\left(V_H^{St}\right) = M_H$. ☐

Taking advantage of Theorem 3.2.4, it is straightforward to deduce from (3.50) that $\operatorname{div} \boldsymbol{u}_H = 0$ in $T\cap\Omega^\varepsilon$ for $T \in \mathcal{T}_H$. We can eliminate pressure from (3.49)–(3.50) by introducing a subspace of V_H^{St}:

$$Z_H^{St} = \left\{\boldsymbol{v} \in V_H^{St} \text{ such that } \operatorname{div} \boldsymbol{v} = 0,\ \forall T \in \mathcal{T}_H\right\}$$

As a result, (3.49)–(3.50) is equivalent to

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \boldsymbol{f}\cdot\boldsymbol{v},\ \forall\boldsymbol{v}\in Z_H^{St} \tag{3.55}$$

The existence and uniqueness of a solution $\boldsymbol{u}_H$ to this equation is guaranteed by Theorem 2.1.2. Then Theorem 3.2.4 implies that there exists one and only one pressure $\bar{p}_H \in M_H$.

Therefore we have proved the existence and uniqueness of a solution $(\boldsymbol{u}_H, \bar{p}_H)$ to the problem (3.49)–(3.50).

3.2.4.2 The discrete variational formulation of the Navier-Stokes problem

Similarly, the discrete variational formulation of Navier-Stokes problem (1.1) is: find the $(\boldsymbol{u}_H, p_H) \in X_H^{St}$ such that

$$c_H^{NS}\left((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)\right) = (\boldsymbol{f}, \boldsymbol{v}), \ \forall (\boldsymbol{v}, q) \in X_H^{St} \tag{3.56}$$

where

$$\begin{aligned} c_H^{NS}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = &\sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \rho \left(\boldsymbol{u}_H \cdot \nabla\right) \boldsymbol{u}_H \cdot \boldsymbol{v}\right) \\ &+ \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \left(-p_H \operatorname{div} \boldsymbol{v} - q \operatorname{div} \boldsymbol{u}_H\right) \end{aligned}$$

Using the same arguments as for Stokes problems, the discrete variational formulation (3.56) is equivalent to: find $\boldsymbol{u}_H \in V_H^{St}$ and $\bar{p}_H \in M_H$ such that

$$a_H^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) + b_H\left(\boldsymbol{v}, \bar{p}_H\right) = F_H\left(\boldsymbol{v}\right), \ \forall \boldsymbol{v} \in V_H^{St} \tag{3.57}$$

$$b_H\left(\boldsymbol{u}_H, q\right) = 0, \ \forall q \in M_H \tag{3.58}$$

where

$$a_H^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) = \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \rho \left(\boldsymbol{u}_H \cdot \nabla\right) \boldsymbol{u}_H \cdot \boldsymbol{v}\right)$$

and the forms $b_H(.,.)$ and $F_H(.,.)$ are the same as (3.52)–(3.53).

The formulation (3.57)–(3.58) is the Galerkin formulation of the Navier-Stokes problem (1.1) on the space V_H^{St}. It is known that this formulation is unstable in the convection-dominated regime and oscillations appear due to the central-difference type approximation of the convection term. One can severely refine the mesh to eliminate the oscillations, so that the convection no longer dominates on an element level. However, multiscale finite element methods are developed to capture macroscopic features of the problem on a rather coarse mesh. It is thus contradictory to refine severely the coarse-scale mesh only to prevent oscillations. We propose thus an alternative to the Galerkin formulation to preclude oscillations without refining the mesh.

In the FE framework, [33] has proposed a *Streamline Upwind/Petrov-Galerkin* (SUPG) formulation for convection dominated problems. Extended to a Petrov-Galerkin formulation, the standard Galerkin weighting functions are modified by adding a streamline upwind perturbation, which acts only in the flow direction, a priori eliminating the possibility of any crosswind diffusion. The modified weighting function can be applied to all terms in the equation, resulting in a strongly consistent weighted residual formulation.

Therefore, we are motivated to develop a stabilized version of (3.57)–(3.58) by adapting the original SUPG method to the multiscale context. However, we point out that the direct application of SUPG stabilization on multiscale finite element methods leads to an approach that is not strongly consistent, because the basis functions are not known analytically but only up to the numerical error present in the offline computation. However, the lack of consistency is minimal and the method we develop still works well in numerical experiments.

In the SUPG formulation, the test function $\tilde{\boldsymbol{v}}$ is defined by:

$$\tilde{\boldsymbol{v}} = \boldsymbol{v} + \tau \boldsymbol{u}_H \cdot \nabla \boldsymbol{v}$$

where $\boldsymbol{v}$ is the Galerkin-type test function, $\boldsymbol{u}_H$ is the convective velocity field and τ is the stabilization coefficient. τ has the dimension of time and it is a function of element parameters such as element dimension and element Peclet number. In this work, we choose a τ_T which converges to 0 in the limit of $H \to 0$.

Applying the SUPG stabilization method to the whole Navier-Stokes equations yields: find $(\boldsymbol{u}_H, p_H) \in X_H^{St}$ such that

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(-\mu\Delta\boldsymbol{u}_H + \rho\left(\boldsymbol{u}_H\cdot\nabla\right)\boldsymbol{u}_H + \nabla p_H\right)\cdot\left(\boldsymbol{v} + \tau_T\boldsymbol{u}_H\cdot\nabla\boldsymbol{v}\right) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \boldsymbol{f}\cdot\left(\boldsymbol{v} + \tau_T\boldsymbol{u}_H\cdot\nabla\boldsymbol{v}\right) \tag{3.59}$$

where τ_T is the stabilization coefficient defined on T.

Comparing (3.59) and (3.57), the additional terms are the stabilization terms. In the left hand side of (3.59), the stabilization term is noted as $a_{H,stab}^{NS}$:

$$a_{H,stab}^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(-\mu\Delta\boldsymbol{u}_H + \rho\left(\boldsymbol{u}_H\cdot\nabla\right)\boldsymbol{u}_H + \nabla p_H\right)\cdot\left(\tau_T\boldsymbol{u}_H\cdot\nabla\boldsymbol{v}\right)$$

Substituting $p_H = \bar{p}_H + \pi_H\left(\boldsymbol{u}_H\right)$ into this equation, we obtain

$$a_{H,stab}^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(-\mu\Delta\boldsymbol{u}_H + \rho\left(\boldsymbol{u}_H\cdot\nabla\right)\boldsymbol{u}_H + \nabla\pi_H\left(\boldsymbol{u}_H\right) + \nabla\bar{p}_H\right)\cdot\left(\tau_T\boldsymbol{u}_H\cdot\nabla\boldsymbol{v}\right)$$

Besides, definition of V_H^{St} (3.12) implies that

$$\int_{T\cap\Omega^\varepsilon} \left(-\mu\Delta\boldsymbol{u}_H + \nabla\pi_H\left(\boldsymbol{u}_H\right)\right)\cdot\left(\tau_T\boldsymbol{u}_H\cdot\nabla\boldsymbol{v}\right) = 0,\ \forall T\in\mathcal{T}_H$$

Finally the stabilization term $a_{H,stab}^{NS}$ is reduced to

$$a_{H,stab}^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\rho\left(\boldsymbol{u}_H\cdot\nabla\right)\boldsymbol{u}_H + \nabla\bar{p}_H\right)\cdot\left(\tau_T\boldsymbol{u}_H\cdot\nabla\boldsymbol{v}\right)$$

We choose to neglect the stabilization over the gradient term. Hence the stabilized version of the Navier-Stokes problem (3.59) can be written as: find $\boldsymbol{u}_H \in V_H^{St}$, $\bar{p}_H \in M_H$ such that

$$a_H^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) + a_{H,stab}^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) + b_H\left(\boldsymbol{v}, \bar{p}_H\right) = F_H\left(\boldsymbol{v}\right) + F_{H,stab}\left(\boldsymbol{v}\right),\ \forall\boldsymbol{v}\in V_H^{St} \tag{3.60}$$

$$b_H\left(\boldsymbol{u}_H, q\right) = 0,\ \forall q\in M_H \tag{3.61}$$

with stabilization terms

$$a_{H,stab}^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \rho\left(\left(\boldsymbol{u}_H\cdot\nabla\right)\boldsymbol{u}_H\right)\cdot\left(\tau_T\boldsymbol{u}_H\cdot\nabla\boldsymbol{v}\right) \tag{3.62}$$

$$F_{H,stab}\left(\boldsymbol{v}\right) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \boldsymbol{f}\cdot\left(\tau_T\boldsymbol{u}_H\cdot\nabla\boldsymbol{v}\right) \tag{3.63}$$

where

$$\tau_T(\boldsymbol{x}) = \frac{H}{2|\boldsymbol{u}_H(\boldsymbol{x})|}\left[\coth\left(\frac{|\boldsymbol{u}_H(\boldsymbol{x})|H}{2\nu}\right) - \frac{2\nu}{|\boldsymbol{u}_H(\boldsymbol{x})|H}\right] \tag{3.64}$$

where $\nu = \dfrac{\rho}{\mu}$ and H is the element size of the coarse-scale mesh.

Since we choose a τ_T which converges to 0 in the limit of $H \to 0$, system (3.60) is strongly consistent.

Remark. *The choice of an optimal stabilization parameter is a little vague but important, since it can affect the accuracy of the numerical approximation. In practice, when computing τ_T on each coarse element, we use an average of $\boldsymbol{u}_H$ over T instead of using the varying $\boldsymbol{u}_H(\boldsymbol{x})$ for $\boldsymbol{x} \in T$. Consequently, the stabilization coefficient is element-wise constant on the coarse mesh $\mathcal{T}_H$.*

3.2.4.3 The discrete variational formulation of the Oseen problem

The discrete variational formulation of Oseen problem (1.3) is: find $(\boldsymbol{u}_H, p_H) \in X_H^{St}$ such that

$$c_H^{Os}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = (\boldsymbol{f}, \boldsymbol{v}),\ \forall(\boldsymbol{v}, q) \in X_H^{St}$$

where

$$\begin{aligned} c_H^{Os}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = &\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} (\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \rho(\boldsymbol{U}_o\cdot\nabla)\boldsymbol{u}_H\cdot\boldsymbol{v}) \\ &+ \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} (-p_H \operatorname{div}\boldsymbol{v} - q\operatorname{div}\boldsymbol{u}_H) \end{aligned}$$

Similar to stabilized formulations of Navier-Stokes problem (3.60)–(3.61), the stabilized formulation of the Oseen problem is: find $\boldsymbol{u}_H \in V_H^{St}$, $\bar{p}_H \in M_H$ such that

$$a_H^{Os}(\boldsymbol{u}_H, \boldsymbol{v}) + a_{H,stab}^{Os}(\boldsymbol{u}_H, \boldsymbol{v}) + b_H(\boldsymbol{v}, \bar{p}_H) = F_H(\boldsymbol{v}) + F_{H,stab}(\boldsymbol{v}),\ \forall\boldsymbol{v}\in V_H^{St} \tag{3.65}$$

$$b_H(\boldsymbol{u}_H, q) = 0,\ \forall q \in M_H \tag{3.66}$$

where

$$a_H^{Os}(\boldsymbol{u}_H, \boldsymbol{v}) = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} (\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \rho(\boldsymbol{U}_o\cdot\nabla)\boldsymbol{u}_H\cdot\boldsymbol{v}) \tag{3.67}$$

$$a_{H,stab}^{Os}(\boldsymbol{u}_H, \boldsymbol{v}) = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} \rho((\boldsymbol{U}_o\cdot\nabla)\boldsymbol{u}_H)\cdot(\tau_T\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}) \tag{3.68}$$

$$F_{H,stab}(\boldsymbol{v}) = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} \boldsymbol{f}\cdot(\tau_T\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}) \tag{3.69}$$

and the forms $b_H(.,.)$ and $F_H(.,.)$ are the same as (3.52)–(3.53). The stabilization coefficient is defined by

$$\tau_T(\boldsymbol{x}) = \frac{H}{2|\boldsymbol{U}_o(\boldsymbol{x})|}\left[\coth\left(\frac{|\boldsymbol{U}_o(\boldsymbol{x})|H}{2\nu}\right) - \frac{2\nu}{|\boldsymbol{U}_o(\boldsymbol{x})|H}\right] \tag{3.70}$$

where $\nu = \dfrac{\rho}{\mu}$ and H is the element size of the coarse-scale mesh.

In practice, when computing τ_T on each coarse element, we use an average of $\boldsymbol{U}_o$ over T instead of using the varying $\boldsymbol{U}_o(\boldsymbol{x})$ for $\boldsymbol{x} \in T$. Consequently, the stabilization coefficient is element-wise constant on the coarse mesh $\mathcal{T}_H$.

3.2.5 The reconstruction of fine-scale features

After solving the coarse-scale problem, we obtain the coarse-scale solutions $\boldsymbol{u}_H$ and $\bar{p}_H$. Then we reconstruct the fine-scale features of the solution for visualization. On each coarse element $T \in \mathcal{T}_H$, making use of the expression of V_H^{St} defined by (3.40) and the formula of π_H defined by (3.47), the fine-scale velocity and pressure are approximated by

$$\boldsymbol{u}_H|_T = \sum_{E\in\mathcal{E}(T)} \sum_{i=1}^{s} u_{E,i}\boldsymbol{\Phi}_{E,i}, \quad p_H|_T = \sum_{E\in\mathcal{E}(T)} \sum_{i=1}^{s} u_{E,i}\pi_{E,i} + \bar{p}_H|_T$$

The mathematical formulation of fine-scale solutions is very simple. However, in practice, it is much more complicate to implement this step in the software. The implementation of this part will be presented in Chapter 5.

3.3 The Crouzeix-Raviart multiscale finite element method defined by Oseen equations

In this section, we present a Crouzeix-Raviart multiscale finite element method where local problems are defined by Oseen equations. This method was at first proposed in [117] where some numerical experiments with small Oseen velocities $\boldsymbol{U}_o$ were presented. However, a serious flaw of this method is that when $\boldsymbol{U}_o$ becomes important, oscillations appear in the solution of local problems even when applying the Upwind scheme on the Oseen term. Besides, the well-posedness of local problems was not proved. In order to solve this problem, we propose a new definition of local problems and prove their well-posedness. This improvement precludes oscillations in the solution of local problems for whatever values of $\boldsymbol{U}_o$. Besides, we propose to solve Oseen problems (1.3) as well as Navier-Stokes problems (1.1) on the coarse mesh using this multiscale method. The SUPG stabilization technique is developed to stabilize the solution of Navier-Stokes problems.

3.3.1 The construction of the approximation space X_H^{Os}

The Oseen problem is more or less similar to the Stokes problem except the linear Oseen term. We first introduce the space of unresolved fine-scale features X_H^0 as $X_H^0 = V_H^0 \times M_H^0$. We recall the definition of V_H^0 and M_H^0:

$$V_H^0 = \left\{ \boldsymbol{u} \in V_H^{ext} \text{ such that } \int_{E\cap\Omega^\varepsilon} \boldsymbol{u}\cdot\boldsymbol{\omega}_{E,j} = 0 \; \forall E \in \mathcal{E}_H, \; \forall j = 1, \cdots, s \right\} \tag{3.71}$$

$$M_H^0 = \left\{ p \in M \text{ such that } \int_{T\cap\Omega^\varepsilon} p = 0 \; \forall T \in \mathcal{T}_H \right\} \tag{3.72}$$

where weighting functions $\boldsymbol{\omega}_{E,j}$ for $E \in \mathcal{E}_H$ and $j = 1, \cdots, s$ are practically the same as what were defined in (3.7)–(3.9).

Since this Crouzeix-Raviart multiscale finite element method is defined by Oseen equations, we denote the approximation space X_H by X_H^{Os} in order to distinguish from X_H^{St}. The construction of X_H^{Os} is more or less similar to that of X_H^{St} presented in subsection 3.2.1.

In section 2.3, we introduced two bilinear forms $a^{Os}(.,.)$ and $\widehat{a}^{Os}(.,.)$ in the variational formulation of Oseen problem. These two forms are completely equivalent when the velocity space is included in $\left(H_0^1\left(\Omega^\varepsilon\right)\right)^d$. Similarly, in the discrete form, we introduce two bilinear forms $a_H^{Os}(.,.)$ and $\widehat{a}_H^{Os}(.,.)$:

$$
\begin{aligned}
a_H^{Os}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) &= \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \rho\left(\boldsymbol{U}_o \cdot \nabla\right) \boldsymbol{u}_H \cdot \boldsymbol{v}\right) \\
\widehat{a}_H^{Os}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) &= \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla\right) \boldsymbol{u}_H \cdot \boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla\right) \boldsymbol{v} \cdot \boldsymbol{u}_H\right) \\
&\quad + \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(-\frac{1}{2}\rho \boldsymbol{u}_H \boldsymbol{v} \text{ div } \boldsymbol{U}_o\right)
\end{aligned}
$$

which all involve broken integral since the approximation space of velocity is not included in $\left(H_0^1\left(\Omega^\varepsilon\right)\right)^d$.

On the broken space

$$
\left\{\boldsymbol{v} \in \left(L^2\left(\Omega^\varepsilon\right)\right)^d,\ \boldsymbol{v} \in \left(H^1\left(T \cap \Omega^\varepsilon\right)\right)^d \text{ for any } T \in \mathcal{T}_H,\ \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial\Omega^\varepsilon\right\}
$$

and under the classical assumption that div $\boldsymbol{U}_o \leq 0$, it is easy to see that

$$
\begin{aligned}
\widehat{a}_H^{Os}\left(\boldsymbol{v}, \boldsymbol{v}\right) &= \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{v} : \nabla \boldsymbol{v} - \frac{1}{2}\rho \boldsymbol{v}\boldsymbol{v} \text{ div } \boldsymbol{U}_o\right) \\
&\geq \sum_{T\in\mathcal{T}_H} \mu \left\|\nabla \boldsymbol{v}\right\|^2_{L^2(T\cap\Omega^\varepsilon)}
\end{aligned}
$$

under some small additional constraints on the broken space (such as weak continuity of functions across element edges), we prove that $\widehat{a}_H^{Os}(.,.)$ is coercive. However, it is more difficult to prove that $a_H^{Os}(.,.)$ is coercive. Thus we prefer to use $\widehat{a}_H^{Os}(.,.)$ than $a_H^{Os}(.,.)$ in this work (except in subsection 3.3.5).

We introduce also bilinear forms $c^{Os}\left(.,.\right)$ and $\widehat{c}_H^{Os}(.,.)$:

$$
\begin{aligned}
\widehat{c}_H^{Os}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) &= \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \rho\left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right) \boldsymbol{v}\right) \\
&\quad + \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(-p_H \text{ div } \boldsymbol{v} - q \text{ div } \boldsymbol{u}_H\right) \qquad (3.73) \\
\widehat{c}_H^{Os}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) &= \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right) \boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{v}\right) \boldsymbol{u}_H\right) \\
&\quad + \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(-\frac{1}{2}\rho \boldsymbol{u}\boldsymbol{v} \text{ div } \boldsymbol{U}_o\right) + \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(-p_H \text{ div } \boldsymbol{v} - q \text{ div } \boldsymbol{u}_H\right) \\
&\qquad (3.74)
\end{aligned}
$$

which all involve broken integral. Similarly, $c^{Os}\left(.,.\right)$ and $\widehat{c}_H^{Os}(.,.)$ are completely equivalent if the approximation space of velocity is included in $\left(H_0^1\left(\Omega^\varepsilon\right)\right)^d$. Since this is not the case here, we prefer to use $\widehat{c}_H^{Os}(.,.)$ instead of $c^{Os}\left(.,.\right)$ (except in subsection 3.3.5).

Assumption 3.3.1. *We assume that the Oseen velocity* $\boldsymbol{U}_o$ *verifies that*

$$
\textit{div } \boldsymbol{U}_o = 0 \textit{ in } T \cap \Omega^\varepsilon,\ \forall T \in \mathcal{T}_H
$$

Definition 3.3.1. *The space X_H^{Os} is defined as the "orthogonal" complement of X_H^0 with respect to the bilinear form $\widehat{c}_H^{Os}(.,.)$:*

$$(\boldsymbol{u}_H, p_H) \in X_H^{Os} \Longleftrightarrow \widehat{c}_H^{Os}\left((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)\right) = 0,\ \forall (\boldsymbol{v}, q) \in X_H^0 \tag{3.75}$$

Taking into account Assumption 3.3.1, $\widehat{c}_H^{Os}(.,.)$ defined by (3.74) becomes:

$$\begin{aligned}\widehat{c}_H^{Os}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right)\boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{v}\right)\boldsymbol{u}_H\right) \\ + \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(-p_H \, div\, \boldsymbol{v} - q\, div\, \boldsymbol{u}_H\right)\end{aligned}$$

We recall that in the broken integral above, the div $\boldsymbol{U}_o$ is only defined on each coarse element of $\mathcal{T}_H$.

Definition 3.3.2. *Let the functional spaces $M_H \subset M$ and $V_H^{Os} \subset V_H^{ext}$ be defined by:*

$$M_H = \left\{q \in L_0^2(\Omega) \text{ such that } q\,|_T = \text{const},\ \forall T \in \mathcal{T}_H\right\} \tag{3.76}$$

$$V_H^{Os} = \left\{\begin{array}{l} \boldsymbol{v} \in \left(L^2\left(\Omega^\varepsilon\right)\right)^d : \forall T \in \mathcal{T}_H,\ \exists \zeta^T \in L_0^2\left(T \cap \Omega^\varepsilon\right),\ \exists \alpha^T \in \mathbb{R} \text{ such that} \\ -\mu\Delta\boldsymbol{v} + \rho\left(\boldsymbol{U}_o \cdot \nabla\right)\boldsymbol{v} + \nabla\zeta^T = \boldsymbol{0} \text{ in } T\cap\Omega^\varepsilon \\ div\ \boldsymbol{v} = \alpha^T \text{ in } T\cap\Omega^\varepsilon \\ \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T \\ \mu\nabla\boldsymbol{v}\boldsymbol{n} - \dfrac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\boldsymbol{n}\right)\boldsymbol{v} - \zeta^T\boldsymbol{n} \in \left\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E \in \mathcal{E}\left(T\right) \end{array}\right\} \tag{3.77}$$

where $\mathcal{E}(T)$ is the set of faces composing ∂T.

Definition 3.3.3. *For any $\boldsymbol{v} \in V_H^{Os}$, on any element $T \in \mathcal{T}_H$, definition (3.77) implies that there exists a function ζ^T which is uniquely determined by $\boldsymbol{v}$. We can thus define a linear operator $\pi_H : V_H^{Os} \to M_H^0$ such that for any $\boldsymbol{v} \in V_H^{Os}$, $\pi_H\left(\boldsymbol{v}\right) = \zeta^T$ on any triangle $T \in \mathcal{T}_H$.*

Theorem 3.3.1. *Using M_H (3.76) and V_H^{Os} (3.77), we define $\tilde{X}_H^{Os}$ by*

$$\tilde{X}_H^{Os} = \text{span}\left\{\left(\boldsymbol{u}_H,\ \pi_H\left(\boldsymbol{u}_H\right) + \bar{p}_H\right),\ \boldsymbol{u}_H \in V_H^{Os},\ \bar{p}_H \in M_H\right\} \tag{3.78}$$

then the space X_H^{Os} defined by (3.75) satisfies the following property

$$X_H^{Os} = \tilde{X}_H^{Os} \tag{3.79}$$

This theorem is new and we provide its detailed proof in what follows.

Proof. We first prove that $(\boldsymbol{u}_H, p_H) \in X_H^{Os}$ in the sense of definition (3.75) belongs to the space defined by (3.78). Let $(\boldsymbol{u}_H, p_H) \in X_H^{Os}$ in the sense of definition (3.75), we have $\forall (\boldsymbol{v}, q) \in X_H^0$

$$\begin{aligned}\widehat{c}_H^{Os}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right)\boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{v}\right)\boldsymbol{u}_H\right) \\ + \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(-p_H \text{ div } \boldsymbol{v} - q \text{ div } \boldsymbol{u}_H\right) = 0\end{aligned} \tag{3.80}$$

In a first step, on any element $T \in \mathcal{T}_H$, we define $\bar{p}_H|_T$ the average of pressure p_H on this element. We define $\bar{p}_H$ a function whose restriction on T equals to $\bar{p}_H|_T$ on any element $T \in \mathcal{T}_H$.

It is easy to see that $\bar{p}_H$ is well defined and $\bar{p}_H \in M_H$. Then $p'_H = (p_H - \bar{p}_H) \in M_H^0$ defined by (3.72). Consequently, we can decompose the pressure p_H in a unique way as

$$p_H = \bar{p}_H + p'_H \text{ with } \bar{p}_H \in M_H \text{ and } p'_H \in M_H^0$$

By virtue of this decomposition, the term concerning p_H in (3.80) can be decomposed as

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} p_H \text{ div } \boldsymbol{v} = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} + \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} p'_H \text{ div } \boldsymbol{v} \tag{3.81}$$

Now we compute the first term in the right hand side of (3.81). As $\bar{p}_H$ is constant on $T \in \mathcal{T}_H$, we have

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = \sum_{T\in\mathcal{T}_H} \bar{p}_H|_T \int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{v},\ \forall \boldsymbol{v} \in V_H^0$$

Taking advantage of Assumption 3.1.1, for any $\boldsymbol{v} \in V_H^0$, the definition of V_H^0 implies that

$$\int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{v} = \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v}\cdot\boldsymbol{n}$$

As a result, the first term in the right hand side of (3.81) vanishes

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = 0,\ \forall \boldsymbol{v} \in V_H^0$$

and then (3.80) reduces to: $\forall \boldsymbol{v} \in V_H^0, \forall q \in M_H^0$

$$\begin{aligned} \widehat{c}_H^{Os}\left((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)\right) = &\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{u}\right)\boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right)\boldsymbol{u}\right) \\ &- \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} p'_H \text{ div } \boldsymbol{v} - \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0 \end{aligned} \tag{3.82}$$

In a second step, choosing an element $T \in \mathcal{T}_H$ and the test function $\boldsymbol{v} = \boldsymbol{0}$, for any $q \in M_H^0$ with q vanishing outside T, (3.82) becomes

$$\int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0$$

We deduce from this equation that there exists a constant $\alpha^T \in \mathbb{R}$ in $T\cap\Omega^\varepsilon$ such that

$$\text{div } \boldsymbol{u}_H = \alpha^T \text{ in } T\cap\Omega^\varepsilon \tag{3.83}$$

In a third step, we observe that for any face $E \in \mathcal{E}(T)$, there exist some (non unique) functions $\boldsymbol{v}_{E,i} \in \left(H^1\left(T\cap\Omega^\varepsilon\right)\right)^d$, $i = 1, \cdots, s$ such that

$$\begin{cases} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v}_{E,i}\cdot\boldsymbol{\omega}_{F,j} = \delta_{E,F}\delta_{i,j}, \forall F \in \mathcal{E}\left(T\right),\ j = 1, \cdots, s \\ \boldsymbol{v}_{E,i} = \boldsymbol{0},\ \text{on } \partial B^\varepsilon \cap T \end{cases} \tag{3.84}$$

We denote by $V(T)$ the set of functions in $\left(H^1\left(T\cap\Omega^\varepsilon\right)\right)^d$ that vanish on $\partial B^\varepsilon \cap T$:

$$V(T) = \left\{\boldsymbol{v} \in \left(H^1(T\cap\Omega^\varepsilon)\right)^d \text{ such that } \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T\right\}$$

It is easy to check that $V(T)$ can be decomposed as

$$V(T) = V_{\int 0}(T) \oplus \operatorname{span}\{\boldsymbol{v}_{E,i},\ \forall E \in \mathcal{E}(T),\ \forall i = 1, \cdots, s\}$$

where

$$V_{\int 0}(T) = \left\{ \begin{array}{l} \boldsymbol{v} \in \left(H^1(T \cap \Omega^\varepsilon)\right)^d : \int_{E \cap \Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{E,i} = 0,\ \forall E \in \mathcal{E}(T),\ \forall i = 1, \cdots, s \\ \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T \end{array} \right\}$$

By virtue of this decomposition, for any $\boldsymbol{v} \in V(T)$, there exist $\tilde{\boldsymbol{v}} \in V_{\int 0}(T)$ and $\beta_{F,1}, \cdots, \beta_{F,s} \in \mathbb{R}$ such that

$$\boldsymbol{v} = \tilde{\boldsymbol{v}} + \sum_{F \in \mathcal{E}(T)} \sum_{j=1}^{s} \beta_{F,j} \boldsymbol{v}_{F,j} \tag{3.85}$$

Now let us compute the coefficient $\beta_{E,i}$ for any $E \in \mathcal{E}(T)$ and $i = 1, \cdots, s$. Multiplying (3.85) by $\boldsymbol{\omega}_{E,i}$ and integrating over E yields

$$\int_{E \cap \Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{E,i} = \int_{E \cap \Omega^\varepsilon} \tilde{\boldsymbol{v}} \cdot \boldsymbol{\omega}_{E,i} + \sum_{F \in \mathcal{E}(T)} \sum_{j=1}^{s} \beta_{F,j} \int_{E \cap \Omega^\varepsilon} \boldsymbol{v}_{F,j} \cdot \boldsymbol{\omega}_{E,i} \tag{3.86}$$

Since $\tilde{\boldsymbol{v}} \in V_H^0$, definition of V_H^0 (3.71) implies that $\int_E \tilde{\boldsymbol{v}} \cdot \boldsymbol{\omega}_{E,i} = 0$. Besides, it is trivial to deduce from (3.84) that

$$\sum_{F \in \mathcal{E}(T)} \sum_{j=1}^{s} \beta_{F,j} \int_{E \cap \Omega^\varepsilon} \boldsymbol{v}_{F,j} \cdot \boldsymbol{\omega}_{E,i} = \beta_{E,i}$$

Finally (3.86) reduces to

$$\int_{E \cap \Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{E,i} = \beta_{E,i} \tag{3.87}$$

Let $\underline{\tilde{\boldsymbol{v}}}$ be the function defined on Ω^ε and equal to $\tilde{\boldsymbol{v}}$ on $T \cap \Omega^\varepsilon$ and $\boldsymbol{0}$ elsewhere. It is obvious that $\underline{\tilde{\boldsymbol{v}}} \in V_H^0$. Taking $q = 0$, (3.82) implies that

$$\begin{aligned} &\sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \underline{\tilde{\boldsymbol{v}}} + \frac{1}{2} \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H \right) \underline{\tilde{\boldsymbol{v}}} - \frac{1}{2} \rho \left(\boldsymbol{U}_o \cdot \nabla \underline{\tilde{\boldsymbol{v}}} \right) \boldsymbol{u}_H - p'_H \operatorname{div} \underline{\tilde{\boldsymbol{v}}} \right) \\ &= \int_{T \cap \Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \tilde{\boldsymbol{v}} + \frac{1}{2} \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H \right) \tilde{\boldsymbol{v}} - \frac{1}{2} \rho \left(\boldsymbol{U}_o \cdot \nabla \tilde{\boldsymbol{v}} \right) \boldsymbol{u}_H - p'_H \operatorname{div} \tilde{\boldsymbol{v}} \right) = 0 \end{aligned}$$

Substituting $\tilde{\boldsymbol{v}}$ (3.85) into this equation, we obtain

$$\begin{aligned} &\int_{T \cap \Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \frac{1}{2} \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H \right) \boldsymbol{v} - \frac{1}{2} \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{v} \right) \boldsymbol{u}_H - p'_H \operatorname{div} \boldsymbol{v} \right) = \sum_{F \in \mathcal{E}(T)} \sum_{j=1}^{s} \\ &\beta_{F,j} \int_{T \cap \Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v}_{F,j} + \frac{1}{2} \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H \right) \boldsymbol{v}_{F,j} - \frac{1}{2} \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{v}_{F,j} \right) \boldsymbol{u}_H - p'_H \operatorname{div} \boldsymbol{v}_{F,j} \right) \end{aligned} \tag{3.88}$$

Denoting for any $F \in \mathcal{E}(T)$ and for any $j = 1, \cdots, s$

$$\lambda_{F,j} = \int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v}_{F,j} + \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right) \boldsymbol{v}_{F,j} - \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{v}_{F,j}\right) \boldsymbol{u}_H - p'_H \text{ div } \boldsymbol{v}_{F,j}\right)$$

and substituting $\beta_{F,j}$ (3.87), equation (3.88) becomes

$$\int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right) \boldsymbol{v} - \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{v}\right) \boldsymbol{u}_H - p'_H \text{ div } \boldsymbol{v}\right) = \sum_{F\in\mathcal{E}(T)} \sum_{j=1}^{s} \lambda_{F,j} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{F,j} \tag{3.89}$$

On each $T \in \mathcal{T}_H$, making use of Assumption 3.3.1, the Oseen term can be decomposed as

$$\int_{T\cap\Omega^\varepsilon} \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right) \boldsymbol{v} = \int_{T\cap\Omega^\varepsilon} \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right) \boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{v}\right) \boldsymbol{u}_H + \int_{\partial(T\cap\Omega^\varepsilon)} \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}\right) \boldsymbol{u}_H \boldsymbol{v} \tag{3.90}$$

Combining (3.89) and (3.90), equation (3.89) finally becomes

$$\int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right) \boldsymbol{v} - p'_H \text{ div } \boldsymbol{v}\right) = \sum_{F\in\mathcal{E}(T)} \sum_{j=1}^{s} \lambda_{F,j} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{F,j} + \int_{\partial(T\cap\Omega^\varepsilon)} \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}\right) \boldsymbol{u}_H \boldsymbol{v}, \ \forall \boldsymbol{v} \in V(T) \tag{3.91}$$

In (3.91), taking $\boldsymbol{v} = \boldsymbol{0}$ on $\mathcal{E}(T)$ and integrating by parts the left hand side, we obtain

$$-\mu \Delta \boldsymbol{u}_H + \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right) \boldsymbol{v} + \nabla p'_H = \boldsymbol{0} \text{ in } T \cap \Omega^\varepsilon \tag{3.92}$$

Then by writing the variational formulation of (3.92) for $\boldsymbol{v} \in V(T)$ and comparing with (3.89), we deduce that

$$\mu \nabla \boldsymbol{u}_H \boldsymbol{n} - \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}\right) \boldsymbol{u}_H - p'_H \boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\right\} \text{ on } E \cap \Omega^\varepsilon, \ \forall E \in \mathcal{E}(T) \tag{3.93}$$

Finally, combining equations (3.83), (3.25) and (3.93), we obtain the following system:

$$\begin{aligned}
-\mu \Delta \boldsymbol{u}_H + \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right) \boldsymbol{v} + \nabla p'_H &= \boldsymbol{0} \text{ in } T \cap \Omega^\varepsilon \\
\text{div } \boldsymbol{u}_H &= \alpha^T \text{ in } T \cap \Omega^\varepsilon \\
\boldsymbol{u}_H &= \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T \\
\mu \nabla \boldsymbol{u}_H \boldsymbol{n} - \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}\right) \boldsymbol{u}_H - p'_H \boldsymbol{n} &\in \text{span}\left\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\right\} \text{ on } E \cap \Omega^\varepsilon, \ \forall E \in \mathcal{E}(T)
\end{aligned}$$

On on any element $T \in \mathcal{T}_H$, for any $\boldsymbol{u}_H \in V_H^{Os}$ fixed in the formula above, it is easy to see that the gradient $\nabla p'_H$ is uniquely determined by the first equation. Besides, the fact that the average p'_H is 0 over any element implies that p'_H is uniquely determined by $\boldsymbol{u}_H$. Thus we conclude that there exists a linear operator such that $p'_H = \pi_H\left(\boldsymbol{u}_H\right)$ with $\boldsymbol{u}_H \in V_H^{Os}$. We recall that the pressure p_H is decomposed as $p_H = \bar{p}_H + p'_H$. Thus we have proved that $(\boldsymbol{u}_H, p_H)$ defined by (3.75) belongs to the space defined by (3.78).

Reciprocally, we now prove that any $(\boldsymbol{u}_H, p_H) \in \tilde{X}_H^{Os}$ defined by (3.78) satisfies also relation (3.75). Let $(\boldsymbol{u}_H, p_H) \in \tilde{X}_H^{Os}$ defined by (3.78). For any $\boldsymbol{u}_H \in V_H^{Os}$, on any $T \in \mathcal{T}_H$, definition of V_H^{Os} (3.77) implies that there exist a unique $\zeta^T = \pi_H(\boldsymbol{u}_H) \in L_0^2(T \cap \Omega^\varepsilon)$ and a unique $\alpha^T \in \mathbb{R}$ such that

$$-\mu\Delta\boldsymbol{u}_H + \rho(\boldsymbol{U}_o \cdot \nabla\boldsymbol{u}_H)\boldsymbol{v} + \nabla\zeta^T = \boldsymbol{0} \text{ in } T \cap \Omega^\varepsilon \tag{3.94}$$

$$\text{div } \boldsymbol{u}_H = \alpha^T \text{ in } T \cap \Omega^\varepsilon \tag{3.95}$$

$$\boldsymbol{u}_H = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T$$

$$\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - \frac{1}{2}\rho(\boldsymbol{U}_o \cdot \boldsymbol{n})\boldsymbol{u}_H - \zeta^T\boldsymbol{n} \in \text{span}\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\} \text{ on } E \cap \Omega^\varepsilon,\ \forall E \in \mathcal{E}(T) \tag{3.96}$$

For any $\boldsymbol{v} \in V_H^0$, integrating by parts (3.94) yields:

$$\int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \frac{1}{2}\rho(\boldsymbol{U}_o \cdot \nabla\boldsymbol{u}_H)\boldsymbol{v} - \frac{1}{2}\rho(\boldsymbol{U}_o \cdot \nabla\boldsymbol{v})\boldsymbol{u}_H\right) - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = \int_{\partial(T\cap\Omega^\varepsilon)} \left(\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - \frac{1}{2}\rho(\boldsymbol{U}_o \cdot \boldsymbol{n})\boldsymbol{u}_H - \zeta^T\boldsymbol{n}\right) \cdot \boldsymbol{v} \tag{3.97}$$

Equation (3.96) shows that for any $E \in \mathcal{E}(T)$, there exist $\lambda_{E,1}, \cdots, \lambda_{E,s} \in \mathbb{R}$ such that

$$\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - \frac{1}{2}\rho(\boldsymbol{U}_o \cdot \boldsymbol{n})\boldsymbol{u}_H - \zeta^T\boldsymbol{n} = \sum_{i=1}^{s} \lambda_{E,i}\boldsymbol{\omega}_{E,i} \text{ on } E \cap \Omega^\varepsilon$$

Substituting this expression into (3.97), we obtain

$$\int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \frac{1}{2}\rho(\boldsymbol{U}_o \cdot \nabla\boldsymbol{u}_H)\boldsymbol{v} - \frac{1}{2}\rho(\boldsymbol{U}_o \cdot \nabla\boldsymbol{v})\boldsymbol{u}_H\right) - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = \sum_{i=1}^{s} \sum_{E\in\mathcal{E}(T)} \lambda_{E,i} \int_{E\cap\Omega^\varepsilon} \boldsymbol{\omega}_{E,i} \cdot \boldsymbol{v} \tag{3.98}$$

However, for any $\boldsymbol{v} \in V_H^0$, the definition of V_H^0 implies that

$$\sum_{i=1}^{s} \sum_{E\in\mathcal{E}(T)} \lambda_{E,i} \int_{E\cap\Omega^\varepsilon} \boldsymbol{\omega}_{E,i} \cdot \boldsymbol{v} = 0$$

Consequently, (3.98) reduces to

$$\int_{T\cap\Omega^\varepsilon} \left(\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \frac{1}{2}(\boldsymbol{U}_o \cdot \nabla\boldsymbol{u}_H)\boldsymbol{v} - \frac{1}{2}(\boldsymbol{U}_o \cdot \nabla\boldsymbol{v})\boldsymbol{u}_H\right) - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = 0 \tag{3.99}$$

Besides, taking advantage of Assumption 3.1.1, for any $\boldsymbol{v} \in V_H^0$, the divergence theorem and definition of V_H^0 implies that

$$\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = \bar{p}_H|_T \int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{v} = \bar{p}_H|_T \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v} \cdot \boldsymbol{n} = 0 \tag{3.100}$$

Meanwhile, for any $q \in M_H^0$, (3.95) shows that

$$\int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0 \tag{3.101}$$

As a result, summing (3.99), (3.100) and (3.101), we obtain that $\forall\,(\boldsymbol{v},q)\in V_H^0\times M_H^0$,

$$\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v}+\frac{1}{2}\rho\,(\boldsymbol{U}_o\cdot\nabla\boldsymbol{u}_H)\,\boldsymbol{v}-\frac{1}{2}\rho\,(\boldsymbol{U}_o\cdot\nabla\boldsymbol{v})\,\boldsymbol{u}_H\right)-\int_{T\cap\Omega^\varepsilon}\left(\zeta^T+\bar{p}_H|_T\right)\,\mathrm{div}\,\boldsymbol{v}$$
$$-\int_{T\cap\Omega^\varepsilon}q\,\mathrm{div}\,\boldsymbol{u}_H=0$$

Denoting $p_H|_T=\zeta^T+\bar{p}_H|_T$ on each $T\in\mathcal{T}_H$, let p_H be the function which equals to $p_H|_T$ on each $T\in\mathcal{T}_H$, then $(\boldsymbol{u}_H,p_H)\in X_H^{Os}$. Summing this equation on all elements of $\mathcal{T}_H$, we obtain that $\forall\,(\boldsymbol{v},q)\in X_H^0$

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v}+\frac{1}{2}\rho\,(\boldsymbol{U}_o\cdot\nabla\boldsymbol{u}_H)\,\boldsymbol{v}-\frac{1}{2}\rho\,(\boldsymbol{U}_o\cdot\nabla\boldsymbol{v})\,\boldsymbol{u}_H\right)-\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}p_H\,\mathrm{div}\,\boldsymbol{v}$$
$$-\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}q\,\mathrm{div}\,\boldsymbol{u}_H=0$$

which is exactly definition (3.75).

Finally we have proved the identity between (3.78) and (3.75), i.e. property (3.79). □

Theorem 3.3.2. *The space X_H^{ext} can be decomposed as:*

$$X_H^{ext}=X_H^{Os}\oplus X_H^0$$

This theorem is new in this thesis and the proof of this theorem is more or less identical to that of Theorem 3.2.2 and the reader can refer to subsection 3.2.1 for more details.

3.3.2 The local problem defined by Oseen equations

Different to the local problems presented in [117], we now propose a new definition of local problems defined by Oseen equations. Moreover, the well-posedness of the local problem will be proved in subsection 3.3.3.

The strong form For any $E\in\mathcal{E}_H, i=1,\cdots,s$, we construct $\boldsymbol{\Phi}_{E,i}:\Omega^\varepsilon\to\mathbb{R}^d$ and the accompanying pressure $\pi_{E,i}:\Omega^\varepsilon\to\mathbb{R}$ such that $\boldsymbol{\Phi}_{E,i}$ and $\pi_{E,i}$ vanish outside the two coarse elements $T_k\subset\omega_E$, $k\in\{1,2\}$ (only one coarse element if $E\in\partial\Omega$) and solve on T_k:

$$\begin{cases}-\mu\Delta\boldsymbol{\Phi}_{E,i}+\rho\,(\boldsymbol{U}_o\cdot\nabla)\,\boldsymbol{\Phi}_{E,i}+\nabla\pi_{E,i}=\boldsymbol{0}\text{ in }T_k\cap\Omega^\varepsilon,\\ \mathrm{div}\,\boldsymbol{\Phi}_{E,i}=\alpha_{E,i}\text{ in }T_k\cap\Omega^\varepsilon,\\ \mu\nabla\boldsymbol{\Phi}_{E,i}\boldsymbol{n}-\frac{1}{2}\rho\,(\boldsymbol{U}_o\cdot\boldsymbol{n})\,\boldsymbol{\Phi}_{E,i}-\pi_{E,i}\boldsymbol{n}\in\mathrm{span}\,\{\boldsymbol{\omega}_{F,1},\cdots,\boldsymbol{\omega}_{F,s}\}\text{ on }F\cap\Omega^\varepsilon,\ \forall F\in\mathcal{E}(T_k),\\ \boldsymbol{\Phi}_{E,i}=\boldsymbol{0}\text{ on }\partial B^\varepsilon\cap T_k,\\ \int_{F\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\omega}_{F,j}=\begin{cases}\delta_{ij}, & F=E\\ 0, & F\neq E\end{cases}\quad\forall F\in\mathcal{E}\left(T_k\right)\ j=1,\cdots,s.\\ \int_{T_k\cap\Omega^\varepsilon}\pi_{E,i}=0.\end{cases}\tag{3.102}$$

where $\mathcal{E}(T_k)$ is the set of faces of T_k. The constant $\alpha_{E,i}$ depends on T_k and satisfies $\int_{T_k\cap\Omega^\varepsilon}\alpha_{E,i}=\int_{\partial(T_k\cap\Omega^\varepsilon)}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{n}$.

The weak form The weak form of system (3.102) reads: for any $E\in\mathcal{E}\left(T_k\right),\ i=1,\cdots,s$, on the coarse element $T_k\subset\omega_E$ for $k\in\{1,2\}$ (only one coarse element if $E\in\partial\Omega$), find

$\boldsymbol{\Phi}_{E,i} \in \left(H^1(T_k \cap \Omega^\varepsilon)\right)^d$ such that $\boldsymbol{\Phi}_{E,i} = \mathbf{0}$ on $\partial B^\varepsilon \cap T_k$, $\pi_{E,i} \in L_0^2\left(T_k \cap \Omega^\varepsilon\right)$ and Lagrange multipliers $\lambda_{F,1}, \cdots, \lambda_{F,s} \in \mathbb{R}$ for $F \in \mathcal{E}(T_k)$ such that

$$\begin{aligned}
\int_{T_k\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{\Phi}_{E,i} : \nabla\boldsymbol{v} + \int_{T_k\cap\Omega^\varepsilon} \left(\frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{v}\cdot\boldsymbol{\Phi}_{E,i}\right)& \\
- \int_{T_k\cap\Omega^\varepsilon} \pi_{E,i}\,\mathrm{div}\,\boldsymbol{v} + \sum_{F\in\mathcal{E}(T_k)}\sum_{j=1}^{s}\lambda_{F,j}\int_{F\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{F,j} &= 0, \\
\int_{T_k\cap\Omega^\varepsilon} q\,\mathrm{div}\,\boldsymbol{\Phi}_{E,i} &= 0, \\
\sum_{F\in\mathcal{E}(T_k)}\sum_{j=1}^{s}\mu_{F,j}\int_{F\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\omega}_{F,j} &= \mu_{E,i}
\end{aligned}$$

for all $\boldsymbol{v} \in \left(H^1\left(T_k \cap \Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{v} = 0$ on $\partial B^\varepsilon \cap T_k$, $q \in L_0^2\left(T_k \cap \Omega^\varepsilon\right)$, $\mu_{F,j} \in \mathbb{R}$ for all $F \in \mathcal{E}\left(T_k\right)$ and $j = 1, \cdots, s$.

3.3.3 The proof of the well-posedness of the local problem

The local problem defined by Oseen equations (3.102) is different to that defined in [117]. Now we provide two different methods to prove the well-posedness of the local problem.

Choosing a coarse element $K \in \mathcal{T}_H$, let $\mathcal{T}_h\left(K\right)$ be a fine discretization of the domain $K\cap\Omega^\varepsilon$. Let n_K be the number of boundaries composing ∂K and $n_s = n_K \times s$ be the dimension of the vector containing Lagrange multipliers $\left(\lambda_{E,i}^K\right)$ for all $E \in \mathcal{E}(K)$ and $i = 1, \cdots, s$. The local problem (3.102) is solved on the fine mesh $\mathcal{T}_h\left(K\right)$. We introduce the spaces of velocity and pressure:

$$\begin{aligned}
V_K &= \left\{\boldsymbol{v} \in \left(H^1\left(K\cap\Omega^\varepsilon\right)\right)^d,\ \boldsymbol{v} = \mathbf{0} \text{ on } \partial B^\varepsilon \cap K\right\} \\
M_K &= L_0^2\left(K\cap\Omega^\varepsilon\right)
\end{aligned}$$

The weak form of (3.102) is: find $\left(\boldsymbol{u}, p, \left(\lambda_{E,i}^K\right)\right) \in V_K \times M_K \times \mathbb{R}^{n_s}$ such that

$$\widehat{a}_K\left(\boldsymbol{u},\boldsymbol{v}\right) + b_K\left(\boldsymbol{v},p\right) + c_E\left(\boldsymbol{v},\left(\lambda_{E,i}^K\right)\right) = 0,\ \forall\boldsymbol{v}\in V_K \tag{3.103}$$

$$b_K\left(\boldsymbol{u},q\right) = 0,\ \forall q\in M_K \tag{3.104}$$

$$c_E\left(\boldsymbol{u},\left(\mu_{E,i}^K\right)\right) = -\delta_{E,i}^K\mu_{E,i}^K,\ \forall\mu_{E,i}^K\in\mathbb{R},\ \forall E\in\mathcal{E}_H,\ \forall i = 1,\cdots,s \tag{3.105}$$

where

$$\begin{aligned}
\widehat{a}_K\left(\boldsymbol{u},\boldsymbol{v}\right) &= \int_{K\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u} : \nabla\boldsymbol{v} + \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{u}\right)\boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right)\boldsymbol{u}\right) \\
b_K\left(\boldsymbol{v},p\right) &= \int_{K\cap\Omega^\varepsilon} p\,\mathrm{div}\,\boldsymbol{v} \\
c_E\left(\boldsymbol{v},\left(\lambda_{E,i}^K\right)\right) &= \sum_{E\in\mathcal{E}(K)}\sum_{i=1}^{s}\lambda_{E,i}^K\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{E,i}
\end{aligned}$$

We recall that $\delta_{E,i}^K = 0$ or 1 and $\mathrm{div}\,\boldsymbol{U}_o = 0$ according to Assumption 3.3.1.

We introduce the space W_K

$$W_K = \left\{ \boldsymbol{v} \in V_K,\ \int_{E\cap\Omega^\varepsilon} \boldsymbol{v} = 0 \text{ for some } E \in \mathcal{E}\left(T\right) \right\}$$

We prove that the bilinear form $\widehat{a}_K(.,.)$ is coercive on W_K with the help of the Poincaré type inequality proposed in the Lemma 9 of [103]. For any $\boldsymbol{v} \in W_K$, we have

$$\begin{aligned} \widehat{a}_K\left(\boldsymbol{v},\boldsymbol{v}\right) &= \int_{K\cap\Omega^\varepsilon} \mu \nabla\boldsymbol{v} : \nabla\boldsymbol{v} \\ &= \int_{K\cap\Omega^\varepsilon} \mu |\nabla\boldsymbol{v}|^2 \end{aligned}$$

Since the semi-norm $\|\nabla\boldsymbol{v}\|_{L^2(K\cap\Omega^\varepsilon)}$ is equivalent to the full H^1 norm by Poincaré inequality (2.6), we obtain that $\widehat{a}_K(.,.)$ is coercive on W_K.

System (3.103)–(3.105) is called a twofold saddle point problem in the literature and can be viewed as a single saddle point problem defined on $V_K \times (M_K \times \mathbb{R}^{n_s})$. By introducing the following bilinear form

$$\forall \boldsymbol{v} \in V_K,\ \tilde{b}_K\left(\boldsymbol{v}, \left(p, \left(\lambda^K_{E,i}\right)\right)\right) = b_K\left(\boldsymbol{v},p\right) + c_E\left(\boldsymbol{v}, \left(\lambda^K_{E,i}\right)\right)$$

system (3.103)–(3.105) can be reformulated as

$$\widehat{a}_K\left(\boldsymbol{u},\boldsymbol{v}\right) + \tilde{b}_K\left(\boldsymbol{v}, \left(p, \left(\lambda^K_{E,i}\right)\right)\right) = 0,\ \forall \boldsymbol{v} \in V_K \tag{3.106}$$

$$\tilde{b}_K\left(\boldsymbol{u}, \left(q, \left(\mu^K_{E,i}\right)\right)\right) = G^K,\ \forall \left(q, \left(\mu^K_{E,i}\right)\right) \in M_K \times \mathbb{R}^{n_s} \tag{3.107}$$

where $G^K = \left(0, \delta^K_{E,i}\mu^K_{E,i}\right)$.

Now we can apply Theorem 2.1.4 to prove the well-posedness of (3.106)–(3.107). It is easy to check that most of the hypothesis in Theorem 2.1.4 are verified and it remains only to prove that the bilinear form $\tilde{b}_K(.,.)$ satisfies the following inf-sup condition: there exists $\alpha_h > 0$ such that

$$\sup_{\boldsymbol{v}\in V_K} \frac{\tilde{b}_K\left(\boldsymbol{v}, \left(p, \left(\lambda^K_{E,i}\right)\right)\right)}{\left(\|p\|_{L^2(K\cap\Omega^\varepsilon)} + \left\|\left(\lambda^K_{E,i}\right)\right\|\right)\|\boldsymbol{v}\|_{H^1(K\cap\Omega^\varepsilon)}} \geq \alpha_h,\ \forall \left(p, \left(\lambda^K_{E,i}\right)\right) \in M_K \times \mathbb{R}^{n_s} \tag{3.108}$$

Now we check if the bilinear form $\tilde{b}_K(.,.)$ satisfies this inf-sup condition. It is shown in [67] that for any $p \in L^2_0\left(K\cap\Omega^\varepsilon\right)$, there exist a $\boldsymbol{v}_0 \in H^1_0\left(K\cap\Omega^\varepsilon\right)$ and a constant $\alpha_1 > 0$ such that

$$\operatorname{div} \boldsymbol{v}_0 = p \text{ and } \|\boldsymbol{v}_0\|_{H^1(K\cap\Omega^\varepsilon)} \leq \alpha_1 \|p\|_{L^2(K\cap\Omega^\varepsilon)} \tag{3.109}$$

Then we construct a $\boldsymbol{v}$ as

$$\boldsymbol{v} = \sum_{F\in\mathcal{E}(K)} \sum_{j=1}^{s} \lambda^K_{F,j} \boldsymbol{\Phi}^{St}_{F,j} + \boldsymbol{v}_0$$

where $\boldsymbol{\Phi}^{St}_{F,j}$ is the solution of the local problem defined by Stokes equations (3.39). It is trivial to verify that there exists a $\alpha_2 > 0$ such that $\left\|\boldsymbol{\Phi}^{St}_{F,j}\right\|_{H^1(K\cap\Omega^\varepsilon)} \leq \alpha_2$ for any $F \in \mathcal{E}(K)$ and $j = 1, \cdots, s$.

Then we observe that $\boldsymbol{v} \in V_K$ and verifies the following inequality

$$\begin{aligned}\|\boldsymbol{v}\|_{H^1(K\cap\Omega^\varepsilon)} &\leq \|\boldsymbol{v}_0\|_{H^1(K\cap\Omega^\varepsilon)} + \sum_{F\in\mathcal{E}(K)}\sum_{j=1}^{s} \left\|\boldsymbol{\Phi}_{F,j}^{St}\right\|_{H^1(K\cap\Omega^\varepsilon)} \left|\lambda_{F,j}^K\right| \\ &\leq \alpha_1 \|p\|_{L^2(K\cap\Omega^\varepsilon)} + \alpha_2 \left\|\left(\lambda_{F,j}^K\right)\right\|\end{aligned}$$

Substituting $\boldsymbol{v}$ into the bilinear form $\tilde{b}_K(.,.)$, we obtain

$$\begin{aligned}\tilde{b}_K\left(\boldsymbol{v},\left(p,\left(\lambda_{E,i}^K\right)\right)\right) = \int_{K\cap\Omega^\varepsilon}\left(p^2 + p\sum_{F\in\mathcal{E}(K)}\sum_{j=1}^{s}\lambda_{F,j}^K \operatorname{div}\boldsymbol{\Phi}_{F,j}^{St}\right) + \\ \sum_{E\in\mathcal{E}(K)}\sum_{i=1}^{s}\lambda_{E,i}^K\int_{E\cap\Omega^\varepsilon}\left(\boldsymbol{v}_0\cdot\boldsymbol{\omega}_{E,i} + \sum_{F\in\mathcal{E}(K)}\sum_{j=1}^{s}\lambda_{F,j}^K\boldsymbol{\Phi}_{F,j}^{St}\cdot\boldsymbol{\omega}_{E,i}\right)\end{aligned} \tag{3.110}$$

In the first term of the right hand side of (3.110), as div $\boldsymbol{\Phi}_{F,j}^{St}$ is constant on K and $p \in L_0^2(K\cap\Omega^\varepsilon)$, we deduce that

$$\int_{K\cap\Omega^\varepsilon} p\sum_{F\in\mathcal{E}(K)}\sum_{j=1}^{s}\lambda_{F,j}^K \operatorname{div}\boldsymbol{\Phi}_{F,j}^{St} = \sum_{F\in\mathcal{E}(K)}\sum_{j=1}^{s}\lambda_{F,j}^K \operatorname{div}\boldsymbol{\Phi}_{F,j}^{St}\int_{K\cap\Omega^\varepsilon} p = 0$$

In the second term of the right hand side of (3.110), the fact that $\boldsymbol{v}_0 \in H_0^1(K\cap\Omega^\varepsilon)$ implies

$$\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}_0\cdot\boldsymbol{\omega}_{E,i} = 0$$

Besides, it is simple to deduce from (3.105) that

$$\sum_{E\in\mathcal{E}(K)}\sum_{i=1}^{s}\lambda_{E,i}^K\int_{E\cap\Omega^\varepsilon}\left(\sum_{F\in\mathcal{E}(K)}\sum_{j=1}^{s}\lambda_{F,j}^K\boldsymbol{\Phi}_{F,j}^{St}\cdot\boldsymbol{\omega}_{E,i}\right) = \sum_{E\in\mathcal{E}(K)}\sum_{i=1}^{s}\left(\lambda_{E,i}^K\right)^2$$

Combining the results above, (3.110) reduces to

$$\tilde{b}_K\left(\boldsymbol{v},\left(p,\left(\lambda_{E,i}^K\right)\right)\right) = \int_{K\cap\Omega^\varepsilon}p^2 + \sum_{E\in\mathcal{E}(K)}\sum_{i=1}^{s}\left(\lambda_{E,i}^K\right)^2 = \|p\|_{L^2(K\cap\Omega^\varepsilon)}^2 + \left\|\left(\lambda_{E,i}^K\right)\right\|^2$$

By denoting $\tilde{\alpha} = \max(\alpha_1,\alpha_2)$, it is trivial to check that

$$\frac{\tilde{b}_K\left(\boldsymbol{v},\left(p,\left(\lambda_{E,i}^K\right)\right)\right)}{\left(\|p\|_{L^2(K\cap\Omega^\varepsilon)} + \left\|\left(\lambda_{E,i}^K\right)\right\|\right)\|\boldsymbol{v}\|_{H^1(K\cap\Omega^\varepsilon)}} \geq \frac{\|p\|_{L^2(K\cap\Omega^\varepsilon)}^2 + \left\|\left(\lambda_{E,i}^K\right)\right\|^2}{\tilde{\alpha}\left(\|p\|_{L^2(K\cap\Omega^\varepsilon)} + \left\|\left(\lambda_{E,i}^K\right)\right\|\right)^2} \geq \frac{1}{2\tilde{\alpha}}$$

This inequality shows that the inf-sup condition (3.108) is verified with $\alpha_h = \frac{1}{2\tilde{\alpha}}$. Hence we conclude that the problem (3.102) is well-posed.

The work of [90] proposes an alternative to prove the well-posedness of (3.106)–(3.107). We first introduce a subspace of V_K:

$$Z_K = \{\boldsymbol{v}\in V_K \mid b_K(\boldsymbol{v},q) = 0,\ \forall q\in M_K\} \subset V_K$$

It is shown in [90] that proving the inf-sup condition (3.108) is equivalent to proving the following statement: there exists a $\beta_h > 0$ such that

$$\sup_{\boldsymbol{v}\in V_K} \frac{b_K(\boldsymbol{v},p)}{\|p\|_{L^2(K\cap\Omega^\varepsilon)}\ \|\boldsymbol{v}\|_{H^1(K\cap\Omega^\varepsilon)}} \geq \beta_h,\ \forall p \in M_K \tag{3.111}$$

$$\sup_{\boldsymbol{v}\in Z_K} \frac{c_E\left(\boldsymbol{v},\left(\lambda^K_{E,i}\right)\right)}{\left\|(\lambda^K_{E,i})\right\|\ \|\boldsymbol{v}\|_{H^1(K\cap\Omega^\varepsilon)}} \geq \beta_h,\ \forall(\lambda^K_{E,i}) \in \mathbb{R}^{n_s} \tag{3.112}$$

As presented previously, for any $p \in L^2_0(K\cap\Omega^\varepsilon)$, there exist a $\boldsymbol{v}_0 \in H^0_1(K\cap\Omega^\varepsilon)$ and a constant $\beta_1 > 0$ such that

$$\text{div}\ \boldsymbol{v}_0 = p \text{ and } \|\boldsymbol{v}_0\|_{H^1(K\cap\Omega^\varepsilon)} \leq \beta_1 \|p\|_{L^2(K\cap\Omega^\varepsilon)}$$

Thus in a first step, we choose $\boldsymbol{v} = \boldsymbol{v}_0$ and it is easy to check that $\boldsymbol{v}_0 \in V_K$ and

$$\frac{b_K(\boldsymbol{v},p)}{\|p\|_{L^2(K\cap\Omega^\varepsilon)}\ \|\boldsymbol{v}\|_{H^1(K\cap\Omega^\varepsilon)}} \geq \frac{\|p\|^2_{L^2(K\cap\Omega^\varepsilon)}}{\beta_1\|p\|^2_{L^2(K\cap\Omega^\varepsilon)}} = \frac{1}{\beta_1}$$

In a second step, we construct $\boldsymbol{v}$ as

$$\boldsymbol{v} = \sum_{E\in\mathcal{E}(K)}\sum_{i=1}^{s} \lambda^K_{E,i}\boldsymbol{\Phi}^{St}_{E,i}$$

where $\boldsymbol{\Phi}^{St}_{E,i}$ is the solution of (3.39) and there exists a $\beta_2 \in \mathbb{R}$ such that $\left\|\boldsymbol{\Phi}^{St}_{E,i}\right\| \leq \beta_2$. Thus we observe that $\boldsymbol{v} \in Z_K$ and $\|\boldsymbol{v}\|_{H^1(K\cap\Omega^\varepsilon)} \leq \beta_2\left\|\left(\lambda^K_{E,i}\right)\right\|$.

Using these properties, the following inequality is straightforward

$$\frac{c_E\left(\boldsymbol{v},\left(\lambda^K_{E,i}\right)\right)}{\left\|(\lambda^K_{E,i})\right\|\ \|\boldsymbol{v}\|_{H^1(K\cap\Omega^\varepsilon)}} \geq \frac{\left\|(\lambda^K_{E,i})\right\|^2}{\beta_2\left\|(\lambda^K_{E,i})\right\|^2} = \frac{1}{\beta_2}$$

Taking $\beta_h = \min(\frac{1}{\beta_1},\frac{1}{\beta_2})$, we conclude that the inf-sup conditions (3.111)–(3.111) are satisfied. As a result, we have proved that the problem (3.102) is well-posed.

3.3.4 The basis functions of the space V^{Os}_H

Theorem 3.3.3. *The functions $\boldsymbol{\Phi}_{E,i}$, for $E \in \mathcal{E}_H$ and $i = 1,\cdots,s$ constructed in (3.102) form a basis of the space V^{Os}_H defined in (3.77). In other words,*

$$V^{Os}_H = \text{span}\{\boldsymbol{\Phi}_{E,i},\ E\in\mathcal{E}_H,\ i=1,\cdots,s\}. \tag{3.113}$$

The proof of this theorem is completely similar to that of Theorem 3.2.3 and thus we will not repeat it here.

Consequently, any function of V^{Os}_H can be written with multiscale basis functions as

$$\forall \boldsymbol{u}_H \in V^{Os}_H,\ \boldsymbol{u}_H = \sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s} u_{E,i}\boldsymbol{\Phi}_{E,i}$$

The linear operator π_H (see Definition 3.3.3) can be written explicitly as

$$\forall \boldsymbol{u}_H \in V_H^{Os},\ \pi_H(\boldsymbol{u}_H) = \sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s} u_{E,i}\pi_{E,i} \tag{3.114}$$

3.3.5 The coarse-scale problem and stabilized formulations

In [117], only Oseen problems were solved on the coarse mesh using the approximation space X_H^{Os}. In this thesis, we propose to solve not only Oseen problems, but also Navier-Stokes problems on the coarse mesh using X_H^{Os}.

3.3.5.1 The discrete variational formulation of the Oseen problem

The discrete variational formulation of Oseen problem (1.3) is: find $(\boldsymbol{u}_H, p_H) \in X_H^{Os}$ such that

$$c_H^{Os}\left((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)\right) = (\boldsymbol{f}, \boldsymbol{v}),\ \forall(\boldsymbol{v}, q) \in X_H^{Os} \tag{3.115}$$

where

$$\begin{aligned} c_H^{Os}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H\cdot\boldsymbol{v}\right)\\ + \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(-p_H \operatorname{div}\boldsymbol{v} - q\operatorname{div}\boldsymbol{u}_H\right)\end{aligned}$$

Theorem 3.3.1 shows that p_H can be decomposed as $p_H = \pi_H(\boldsymbol{u}_H) + \bar{p}_H$ with $\bar{p}_H \in M_H$. Besides, it is easy to verify that $(\pi_H(\boldsymbol{u}_H), \operatorname{div}\boldsymbol{v}_H) = 0$ for all $\boldsymbol{u}_H, \boldsymbol{v}_H \in V_H^{Os}$. Making use of this property, (3.115) is equivalent to: find $\boldsymbol{u}_H \in V_H^{Os}$ and $\bar{p}_H \in M_H$ such that

$$a_H^{Os}(\boldsymbol{u}_H, \boldsymbol{v}) + b_H(\boldsymbol{v}, \bar{p}_H) = F_H(\boldsymbol{v}),\ \forall\boldsymbol{v} \in V_H^{Os} \tag{3.116}$$

$$b_H(\boldsymbol{u}_H, q) = 0,\ \forall q \in M_H \tag{3.117}$$

and we recall that

$$\begin{aligned} a_H^{Os}(\boldsymbol{u}_H, \boldsymbol{v}) &= \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H\cdot\boldsymbol{v}\right)\\ \widehat{a}_H^{Os}(\boldsymbol{u}_H, \boldsymbol{v}) &= \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H\cdot\boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{v}\cdot\boldsymbol{u}_H\right)\end{aligned}$$

under Assumption 3.3.1. We recall that $a_H^{Os}(.,.)$ and $\widehat{a}_H^{Os}(.,.)$ are completely equivalent when the approximation space of velocity is in $\left(H_0^1(\Omega^\varepsilon)\right)^d$. Since V_H^{Os} is not included in $\left(H_0^1(\Omega^\varepsilon)\right)^d$, as presented in subsection 3.3.1, it is easier to show that $\widehat{a}_H^{Os}(.,.)$ is coercive.

In this thesis, for error analysis, we compute reference solutions by *TrioCFD* using the finite volume element method. In this software, Neumann boundary condition is defined by $\mu\nabla\boldsymbol{u}_h\boldsymbol{n} - \bar{p}_h\boldsymbol{n} = \boldsymbol{g}$ with $\boldsymbol{u}_h$ and $\bar{p}_h$ the approximated solutions and $\boldsymbol{g}$ a given function. When using the bilinear form $a_H^{Os}(.,.)$ in coarse-scale problems, it is easy to see that the boundary term is $\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - \bar{p}_H\boldsymbol{n} = \boldsymbol{g}$ with $\boldsymbol{g}$ a given function. However, when using the bilinear form $\widehat{a}_H^{Os}(.,.)$ in coarse-scale problems, we observe that the boundary term will be $\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H - \bar{p}_H\boldsymbol{n} = \boldsymbol{g}'$ with $\boldsymbol{g}'$ a given function and $\boldsymbol{g}' \neq \boldsymbol{g}$.

Consequently, in this thesis, we adopt for the bilinear form $a_H^{Os}(.,.)$ instead of $\widehat{a}_H^{Os}(.,.)$ in coarse-scale Oseen problems. The existence and uniqueness of a solution $(\boldsymbol{u}_H, \bar{p}_H)$ to (3.116)–(3.117) can be proved in the same way as in subsection 3.2.4.1.

Stabilized formulation Now we discuss whether the SUPG stabilization method is necessary for the coarse-scale Oseen problem when using the approximation space V_H^{Os}. The stabilized version of the Oseen problem (3.116)–(3.117) is to find $(\boldsymbol{u}_H, p_H) \in X_H^{Os}$ such that

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(-\mu\Delta\boldsymbol{u}_H+\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H+\nabla p_H\right)\cdot\left(\boldsymbol{v}+\tau_T\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right)=\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\boldsymbol{f}\cdot\boldsymbol{v} + \sum_{T\in\mathcal{T}_H}\boldsymbol{f}\cdot\left(\tau_T\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right) \tag{3.118}$$

where τ_T is the stabilization coefficient defined on T. In this work, we choose a τ_T which converges to 0 in the limit of $H \to 0$.

Comparing (3.118) and (3.116), the additional terms in (3.118) are defined as stabilization terms. In the left hand side of (3.118), the stabilization term $a_{H,stab}^{Os}$ is

$$a_{H,stab}^{Os}\left(\boldsymbol{u}_H,\boldsymbol{v}\right)=\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(-\mu\Delta\boldsymbol{u}_H+\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H+\nabla p_H\right)\cdot\left(\tau_T\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right)$$

By substituting $p_H = \pi_H\left(\boldsymbol{u}_H\right) + \bar{p}_H$ into this equation, we obtain

$$a_{H,stab}^{Os}\left(\boldsymbol{u}_H,\boldsymbol{v}\right)= \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(-\mu\Delta\boldsymbol{u}_H+\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H+\nabla \bar{p}_H+\nabla\pi_H\left(\boldsymbol{u}_H\right)\right)\cdot\left(\tau_T\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right)$$

The definition of V_H^{Os} (3.77) implies that:

$$\int_{T\cap\Omega^\varepsilon}\left(-\mu\Delta\boldsymbol{u}_H+\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H+\nabla\pi_H\left(\boldsymbol{u}_H\right)\right)\cdot\left(\tau_T\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}_H\right)=0,\ \forall T\in\mathcal{T}_H$$

Therefore the stabilization term $a_{H,stab}^{Os}$ is reduced to

$$a_{H,stab}^{Os}\left(\boldsymbol{u}_H,\boldsymbol{v}\right)=\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\nabla\bar{p}_H\cdot\left(\tau_T\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}_H\right)$$

We choose to neglect the stabilization over the gradient of pressure. Our numerical simulations reveal that no oscillations appear when solving Oseen problems using V_H^{Os} on the coarse mesh. This confirms that the SUPG stabilization method is not necessary in this case. However, one can also define the SUPG stabilization term by

$$a_{H,stab}^{Os}\left(\boldsymbol{u}_H,\boldsymbol{v}\right)=\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H\cdot\left(\tau_T\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right)$$

We believe that this stabilization term will not significantly affect numerical solutions of Oseen problems on the coarse mesh.

3.3.5.2 The discrete variational formulation of the Navier-Stokes problem

The discrete variational formulation of Navier-Stokes problem (1.1) is: find the $(\boldsymbol{u}_H, p_H) \in X_H^{Os}$ such that

$$c_H^{NS}\left(\left(\boldsymbol{u}_H,p_H\right),\left(\boldsymbol{v},q\right)\right)=\left(\boldsymbol{f},\boldsymbol{v}\right),\ \forall\left(\boldsymbol{v},q\right)\in X_H^{Os} \tag{3.119}$$

Using the same arguments as for the Oseen problem, (3.119) equals to: find $\boldsymbol{u}_H \in V_H^{Os}$ and $\bar{p}_H \in M_H$ such that

$$a_H^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) + b_H\left(\boldsymbol{v}, \bar{p}_H\right) = F_H\left(\boldsymbol{v}\right), \ \forall \boldsymbol{v} \in V_H^{Os} \tag{3.120}$$

$$b_H\left(\boldsymbol{u}_H, q\right) = 0, \ \forall q \in M_H \tag{3.121}$$

In practice, when solving Navier-Stokes problems with Oseen basis functions, the Oseen velocity $\boldsymbol{U}_o$ in local problems (3.102) is unknown. Thus we try to find an approximation of $\boldsymbol{U}_o$ close to the convective velocity of the Navier-Stokes problem. We first solve the Stokes problem (3.49)–(3.50) and then compute an average of the coarse velocity $\boldsymbol{u}_H^{St}$ in each coarse element. This average is then taken as $\boldsymbol{U}_o$ and it is constant in each coarse element.

The stabilized version of (3.120)–(3.121) is: find $\left(\boldsymbol{u}_H, p_H\right) \in X_H^{Os}$ such that

$$a_H^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) + a_{H,stab}^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) + b_H\left(\boldsymbol{v}, \bar{p}_H\right) = F_H\left(\boldsymbol{v}\right) + F_{H,stab}\left(\boldsymbol{v}\right), \ \forall \boldsymbol{v} \in V_H^{Os} \tag{3.122}$$

$$b_H\left(\boldsymbol{u}_H, q\right) = 0, \ \forall q \in M_H \tag{3.123}$$

Comparing equations (3.122)–(3.123) with (3.120)–(3.121), the additional terms are the stabilization terms. The stabilization term $a_{H,stab}^{Ns}$ is defined by

$$a_{H,stab}^{Ns}\left(\boldsymbol{u}_H, \boldsymbol{v}\right) = \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \left(-\mu \Delta \boldsymbol{u}_H + \rho\left(\boldsymbol{u}_H \cdot \nabla\right) \boldsymbol{u}_H + \nabla \bar{p}_H + \nabla \pi_H\left(\boldsymbol{u}_H\right)\right) \cdot \left(\tau_T \boldsymbol{u}_H \cdot \nabla \boldsymbol{v}\right) \tag{3.124}$$

and we choose a τ_T which converges to 0 in the limit of $H \to 0$.

In this work, we choose to neglect the stabilization over $\left(-\mu \Delta \boldsymbol{u}_H + \nabla \bar{p}_H + \nabla \pi_H\left(\boldsymbol{u}_H\right)\right)$. The stabilization term $a_{H,stab}^{NS}$ (3.124) reduces to:

$$a_{H,stab}^{NS}\left(\boldsymbol{u}_H, \boldsymbol{v}_H\right) = \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \left(\rho\left(\boldsymbol{u}_H \cdot \nabla\right) \boldsymbol{u}_H\right) \cdot \left(\tau_T \boldsymbol{u}_H \cdot \nabla \boldsymbol{v}_H\right) \tag{3.125}$$

Moreover, the stabilization term in the right hand side of (3.122) is

$$F_{H,stab}\left(\boldsymbol{v}_H\right) = \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \boldsymbol{f} \cdot \left(\tau_T \boldsymbol{u}_H \cdot \nabla \boldsymbol{v}_H\right) \tag{3.126}$$

where

$$\tau_T\left(\boldsymbol{x}\right) = \frac{H}{2|\boldsymbol{u}_H\left(\boldsymbol{x}\right)|}\left[\coth\left(\frac{|\boldsymbol{u}_H\left(\boldsymbol{x}\right)|H}{2\nu}\right) - \frac{2\nu}{|\boldsymbol{u}_H\left(\boldsymbol{x}\right)|H}\right] \quad \forall \boldsymbol{x} \in T$$

where $\nu = \dfrac{\rho}{\mu}$ and H is the element size of the coarse-scale mesh.

Since we choose a τ_T which converges to 0 in the limit of $H \to 0$, the stabilized formulation (3.122) with stabilization terms (3.125) and (3.126) is consistent. In practice, when computing τ_T on each coarse element, we use an average of $\boldsymbol{u}_H$ over T instead of using the varying $\boldsymbol{u}_H\left(\boldsymbol{x}\right)$ for $\boldsymbol{x} \in T$. Consequently, the stabilization coefficient is element-wise constant on the coarse mesh $\mathcal{T}_H$.

3.3.6 The reconstruction of fine-scale features

After solving the coarse-scale problems, we obtain the coarse-scale solutions $\boldsymbol{u}_H$ and $\bar{p}_H$. Then we reconstruct the fine-scale features of the solution for visualization. On each coarse element

$T \in \mathcal{T}_H$, making use of V_H^{Os} defined by (3.113) and the formula of π_H defined by (3.114), the fine-scale velocity and pressure are approximated by

$$\boldsymbol{u}_H|_T = \sum_{E \in \mathcal{E}(T)} \sum_{i=1}^{s} u_{E,i} \boldsymbol{\Phi}_{E,i}, \quad p_H|_T = \sum_{E \in \mathcal{E}(T)} \sum_{i=1}^{s} u_{E,i} \pi_{E,i} + \bar{p}_H|_T$$

3.4 The Crouzeix-Raviart multiscale finite element method defined by adding solutions of local Stokes and Oseen problems

We presented Crouzeix-Raviart multiscale finite element methods where multiscale basis functions are defined by respectively Stokes equations section 3.2 or Oseen equations section 3.3. Now we investigate whether using the union of these multiscale basis functions can produce more accurate results than using basis functions defined by Stokes equations alone or Oseen equations alone. In this thesis, we propose thus to construct an enriched approximation space of velocity V_H^{St+Os} using V_H^{St} (3.12) and V_H^{Os} (3.77). We recall that V_H^{St} and V_H^{Os} are defined by

$$V_H^{St} = \left\{ \boldsymbol{\Phi}_{E,i}^{St}, \ E \in \mathcal{E}_H, \ i = 1, \cdots, s \right\}$$
$$V_H^{Os} = \left\{ \boldsymbol{\Phi}_{E,i}^{Os}, \ E \in \mathcal{E}_H, \ i = 1, \cdots, s \right\}$$

Definition 3.4.1. *We define the enriched space V_H^{St+Os} by*

$$V_H^{St+Os} = \left\{ \boldsymbol{\Phi}_{E,i}^{St} \in V_H^{St}, \ \boldsymbol{\Phi}_{E,i}^{Os} \in V_H^{Os}, \ E \in \mathcal{E}_H, \ i = 1, \cdots, s \right\}$$

We recall that the approximation space of pressure M_H is defined by:

$$M_H = \left\{ q \in L_0^2(\Omega) \text{ such that } q\,|_T = \text{const}, \ \forall T \in \mathcal{T}_H \right\}$$

By this definition, any function $\boldsymbol{u}_H$ in V_H^{St+Os} can be represented as

$$\boldsymbol{u}_H = \sum_{E \in \mathcal{E}_H} \sum_{i=1}^{s} u_{E,i}^{St} \boldsymbol{\Phi}_{E,i}^{St} + \sum_{E \in \mathcal{E}_H} \sum_{i=1}^{s} u_{E,i}^{Os} \boldsymbol{\Phi}_{E,i}^{Os}$$

We recall the linear operators $\pi_H^{St} : V_H^{St} \to M_H^0$ (see Definition 3.2.3) and $\pi_H^{Os} : V_H^{Os} \to M_H^0$ (see Definition 3.3.3). For any $\boldsymbol{v}_1 \in V_H^{St}$, on any element $T \in \mathcal{T}_H$, there exists a unique ζ_1^T such that $\pi_H^{St}(\boldsymbol{v}_1) = \zeta_1^T$. For any $\boldsymbol{v}_2 \in V_H^{Os}$, on any element $T \in \mathcal{T}_H$, there exists a unique ζ_2^T such that $\pi_H^{Os}(\boldsymbol{v}_2) = \zeta_2^T$. Now we define a new linear operator in what follows.

Definition 3.4.2. *Combining definitions of π_H^{St} and π_H^{Os}, for any $\boldsymbol{w} \in V_H^{St+Os}$, on any element $T \in \mathcal{T}_H$, there exists a ξ^T uniquely determined by $\boldsymbol{w}$. We define a new linear operator $\pi_H^{St+Os} : V_H^{St+Os} \to M_H^0$ such that for any $\boldsymbol{w} \in V_H^{St+Os}$, $\pi_H^{St+Os}(\boldsymbol{w}) = \xi^T$ on any element $T \in \mathcal{T}_H$.*

Proceeding in the same way as in section 3.2 and section 3.3, we can define an explicit formulation of the linear operator π_H^{St+Os}:

$$\forall \boldsymbol{u}_H \in V_H^{St+Os}, \ \pi_H^{St+Os}(\boldsymbol{u}_H) = \sum_{E \in \mathcal{E}_H} \sum_{i=1}^{s} \left(u_{E,i}^{St} \pi_{E,i}^{St} + u_{E,i}^{Os} \pi_{E,i}^{Os} \right)$$

Consequently, the multiscale approximation space is defined by

$$X_H^{St+Os} = \operatorname{span} \left\{ (\boldsymbol{u}_H, \ \pi_H(\boldsymbol{u}_H) + \bar{p}_H), \ \boldsymbol{u}_H \in V_H^{St+Os}, \ \bar{p}_H \in M_H \right\} \tag{3.127}$$

Since the nonlinear convection term in the Navier-Stokes equations increases the difficulty for the solution of the problem, we first try to solve Oseen problems on the coarse mesh using the space V_H^{St+Os} and expect to obtain a better accuracy than using V_H^{St} or V_H^{Os}. The coarse-scale Oseen problem and the stabilized formulation are similar to what were presented in subsection 3.3.5. Numerical experiments are presented in Chapter 6 to investigate the accuracy of this multiscale method.

3.5 The Crouzeix-Raviart multiscale finite element method enriched by bubble functions

In the literature, [54, 102, 105, 113] proposed to enrich the velocity approximation space by bubble functions when solving diffusion or advection-diffusion problems in perforated media. It was shown that the addition of bubble functions is beneficial for the overall accuracy of the Crouzeix-Raviart multiscale finite element methods. For advection-diffusion problems, the difference between [54] and [113] is that [54] uses homogeneous Dirichlet conditions on the boundary of elements whereas [113] imposes the integral type boundary conditions. However, until now, bubble functions are not defined yet for Stokes or Oseen equations in the literature. In this thesis, we propose to define bubble functions by Stokes or Oseen equations in order to improve the accuracy of the multiscale method.

3.5.1 Bubble functions defined by Stokes equations

Our goal is to enrich the space $V_H^{St} = \left\{ \mathbf{\Phi}_{E,i}^{St},\ E \in \mathcal{E}_H,\ i = 1, \cdots, s \right\}$ defined by (3.12) by adding bubble functions defined by Stokes equations. We do not consider the enrichment of V_H^{St} by bubble functions defined by Oseen equations.

We want to decompose the space X_H^{ext} by

$$X_H^{ext} = X_H^{St+b} \oplus X_{H,bubble}^0$$

where X_H^{St+b} is the enriched finite dimensional space consisting of coarse-scale features and $X_{H,bubble}^0$ is the infinite dimensional space consisting of fine-scale features.

Let s be a positive integer and let $\boldsymbol{\omega}_{E,i}$: $E \to \mathbb{R}^d$ be some vector-valued functions associated to $E \in \mathcal{E}_H$ for $i = 1, \cdots, s$. Let r be a positive integer. We associate the vector-valued functions $\boldsymbol{\varphi}_{T,k} : T \to \mathbb{R}^d$ to each coarse element $T \in \mathcal{T}_H$ and $k = 1, \cdots, r$. Then we introduce the space $X_{H,bubble}^0 = V_{H,bubble}^0 \times M_H^0$, where

$$V_{H,bubble}^0 = \left\{ \begin{array}{l} \boldsymbol{u} \in V_H^{ext} \text{ such that } \int_{E \cap \Omega^\varepsilon} \boldsymbol{u} \cdot \boldsymbol{\omega}_{E,j} = 0, \int_{T \cap \Omega^\varepsilon} \boldsymbol{u} \cdot \boldsymbol{\varphi}_{T,k} = 0, \\ \forall T \in \mathcal{T}_H,\ \forall E \in \mathcal{E}_H,\ \forall j = 1, \cdots, s,\ \forall k = 1, \cdots, r \end{array} \right\} \tag{3.128}$$

$$M_H^0 = \left\{ p \in M \text{ such that } \int_{T \cap \Omega^\varepsilon} p = 0\ \forall T \in \mathcal{T}_H \right\} \tag{3.129}$$

Besides, we define $M_H^0(T)$ by

$$M_H^0(T) = L_0^2(T \cap \Omega^\varepsilon)$$

We observe that the space $V_{H,bubble}^0 \neq V_H^0$ defined in (3.5) or (3.71) and the dimension of $V_{H,bubble}^0$ is smaller than that of V_H^0.

Assumption 3.5.1. *In the space $V_{H,bubble}^{0}$, for any $E \in \mathcal{E}_H$, we assume that weighting functions are chosen as*

$$\begin{cases} s = d: \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1, \cdots, \boldsymbol{\omega}_{E,s} = \boldsymbol{e}_d. \\ r = d: \ \boldsymbol{\varphi}_{T,1} = \boldsymbol{e}_1, \cdots, \boldsymbol{\varphi}_{T,r} = \boldsymbol{e}_d. \end{cases}$$

Definition 3.5.1. *The subspace X_H^{St+b} is defined as the "orthogonal" complement of $X_{H,bubble}^{0}$ with respect to the bilinear form $c_H^{St}(.,.)$:*

$$(\boldsymbol{u}_H, p_H) \in X_H^{ext} \iff c_H^{St}\left((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)\right) = 0, \ \forall (\boldsymbol{v}, q) \in X_{H,bubble}^{0} \tag{3.130}$$

where $c_H^{St}(.,.)$ is defined by:

$$c_H^{St}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} - p_H \, div \, \boldsymbol{v} - q \, div \, \boldsymbol{u}_H\right)$$

In [105, 113], Crouzeix-Raviart multiscale finite element methods are enriched by bubble functions defined by diffusion or advection-diffusion equations. Inspired by these work, at the beginning of this study, we propose to construct the following approximation spaces.

Definition 3.5.2. *Let the functional spaces $M_H \subset M$ and $V_H^{St+b} \subset V_H^{ext}$ be defined by*

$$M_H = \left\{q \in L_0^2(\Omega) \ \text{such that} \ q\,|_T = \text{const}, \ \forall T \in \mathcal{T}_H\right\} \tag{3.131}$$

$$V_H^{St+b} = \left\{ \begin{array}{l} \boldsymbol{v} \in \left(L^2(\Omega^\varepsilon)\right)^d : \ \forall T \in \mathcal{T}_H, \ \exists \zeta^T \in M_H^0(T), \ \exists \boldsymbol{\lambda}_T \in \mathbb{R}^d, \\ \exists \alpha^T \in \mathbb{R} \ \text{such that} \\ -\mu \Delta \boldsymbol{v} + \nabla \zeta^T = \boldsymbol{\lambda}_T \ \text{in} \ T \cap \Omega^\varepsilon \\ div \ \boldsymbol{v} = \alpha^T \ \text{in} \ T \cap \Omega^\varepsilon \\ \boldsymbol{v} = \boldsymbol{0} \ \text{on} \ \partial B^\varepsilon \cap T \\ \mu \nabla \boldsymbol{v} \boldsymbol{n} - \zeta^T \boldsymbol{n} \in \{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\} \ \text{on} \ E \cap \Omega^\varepsilon, \ \forall E \in \mathcal{E}(T) \end{array} \right\} \tag{3.132}$$

where $\mathcal{E}(T)$ is the set of faces composing ∂T.

Definition 3.5.3. *For any $\boldsymbol{v} \in V_H^{St+b}$, on any element $T \in \mathcal{T}_H$, definition (3.132) implies that there exists a function ζ^T which is uniquely determined by $\boldsymbol{v}$. We can thus define a linear operator $\pi_H : V_H^{St+b} \to M_H^0$ such that for any $\boldsymbol{v} \in V_H^{St+b}$, $\pi_H(\boldsymbol{v}) = \zeta^T$ on any element $T \in \mathcal{T}_H$.*

Theorem 3.5.1. *With M_H (3.131) and V_H^{St+b} (3.132), we define $\tilde{X}_H^{St+b}$ by*

$$\tilde{X}_H^{St+b} = \text{span}\left\{\left(\boldsymbol{u}_H, \ \pi_H(\boldsymbol{u}_H) + \bar{p}_H\right), \ \boldsymbol{u}_H \in V_H^{St+b}, \ \bar{p}_H \in M_H\right\} \tag{3.133}$$

Then the approximation space X_H^{St+b} defined in (3.130) satisfies the following property:

$$\tilde{X}_H^{St+b} \subset X_H^{St+b} \tag{3.134}$$

Remark. *One could expect that (3.134) would be an equality. We tried to prove the reverse inclusion but unfortunately we were not to to prove it. This proof as well as some remarks are presented specially in subsection 3.5.2.*

Proof. We prove that any $(\boldsymbol{u}_H, p_H) \in \tilde{X}_H^{St+b}$ defined by (3.133) satisfies relation (3.130). Let $(\boldsymbol{u}_H, p_H) \in \tilde{X}_H^{St+b}$ defined by (3.133). For any $\boldsymbol{u}_H \in V_H^{St+b}$, on any $T \in \mathcal{T}_H$, definition of

V_H^{St+b} (3.132) implies that there exist a unique $\zeta^T = \pi_H(\boldsymbol{u}_H) \in M_H^0(T)$, a unique $\boldsymbol{\lambda}_T \in \mathbb{R}^d$ and a unique constant $\alpha^T \in \mathbb{R}$ such that

$$-\mu\Delta\boldsymbol{u}_H + \nabla\zeta^T = \boldsymbol{\lambda}_T \text{ in } T\cap\Omega^\varepsilon \tag{3.135}$$

$$\text{div } \boldsymbol{u}_H = \alpha^T \text{ in } T\cap\Omega^\varepsilon \tag{3.136}$$

$$\boldsymbol{u}_H = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T$$

$$\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - \zeta^T\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{E,i}, \cdots, \boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E \in \mathcal{E}(T) \tag{3.137}$$

For any $\boldsymbol{v} \in V_{H,bubble}^0$, the integration by parts of (3.135) yields:

$$\int_{T\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = \int_{T\cap\Omega^\varepsilon} \boldsymbol{\lambda}_T\cdot\boldsymbol{v} + \int_{\partial(T\cap\Omega^\varepsilon)} \left(\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - \zeta^T\boldsymbol{n}\right)\cdot\boldsymbol{v} \tag{3.138}$$

Equation (3.137) shows that for any $E \in \mathcal{E}(T)$, there exists $\lambda_{E,1}, \cdots, \lambda_{E,s} \in \mathbb{R}$ such that

$$\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - \zeta^T\boldsymbol{n} = \sum_{i=1}^{s} \lambda_{E,i}\boldsymbol{\omega}_{E,i} \text{ on } E\cap\Omega^\varepsilon$$

Substituting this expression into (3.138), we obtain

$$\int_{T\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = \int_{T\cap\Omega^\varepsilon} \boldsymbol{\lambda}_T\cdot\boldsymbol{v} + \sum_{E\in\mathcal{E}(T)}\sum_{i=1}^{s} \lambda_{E,i} \int_E \boldsymbol{\omega}_{E,i}\cdot\boldsymbol{v} \tag{3.139}$$

For any $\boldsymbol{v} \in V_{H,bubble}^0$, definition of $V_{H,bubble}^0$ implies that

$$\sum_{E\in\mathcal{E}(T)}\sum_{i=1}^{s} \lambda_{E,i} \int_E \lambda_{E,i}\cdot\boldsymbol{v} = 0$$

$$\int_{T\cap\Omega^\varepsilon} \boldsymbol{\lambda}_T\cdot\boldsymbol{v} = 0$$

Consequently, (3.139) reduces to:

$$\int_{T\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = 0 \tag{3.140}$$

Besides, making use of Assumption 3.1.1, for any $\boldsymbol{v} \in V_{H,bubble}^0$, the divergence theorem and the definition of $V_{H,bubble}^0$ imply that

$$\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = \bar{p}_H|_T \int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{v} = \bar{p}_H|_T \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v}\cdot\boldsymbol{n} = 0 \tag{3.141}$$

Meanwhile, for any $q \in M_H^0$, (3.136) shows that

$$\int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0 \tag{3.142}$$

As a result, summing (3.140), (3.141) and (3.142), we obtain: for all $(\boldsymbol{v}, q) \in X_{H,bubble}^0$

$$\int_{T\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \left(\zeta^T + \bar{p}_H|_T\right) \text{ div } \boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0$$

Denoting $p_H|_T = \zeta^T + \bar{p}_H|_T$ on each $T \in \mathcal{T}_H$, let p_H be a function which equals to $p_H|_T$ on each $T \in \mathcal{T}_H$, then $(\boldsymbol{u}_H, p_H) \in X_H^{St+b}$. Summing this equation over all elements $T \in \mathcal{T}_H$, we obtain: $\forall\,(\boldsymbol{v}, q) \in X^0_{H,bubble}$

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} p_H \text{ div } \boldsymbol{v} - \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0$$

which is exactly definition (3.130). Consequently, we have proved property (3.134). ☐

3.5.2 Theoretical analysis of the construction of X_H^{St+b}

After having proved (3.134), now let us verify whether $(\boldsymbol{u}_H, p_H) \in X_H^{St+b}$ in the sense of definition (3.130) belongs to the space defined by (3.133). The objective to understand why the addition of bubble functions defined by (3.155) can not improve the accuracy of numerical results and find a remedy to this.

Let $(\boldsymbol{u}_H, p_H) \in X_H^{St+b}$ in the sense of definition (3.130), i.e. $\forall(\boldsymbol{v}, q) \in X^0_{H,bubble}$

$$c_H^{St}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p_H \text{ div } \boldsymbol{v} - q \text{ div } \boldsymbol{u}_H) = 0 \qquad (3.143)$$

In a first step, for any element $T \in \mathcal{T}_H$, denoting $\bar{p}_H|_T$ the average of pressure p_H on this element. We define $\bar{p}_H$ a function whose restriction on T equals to $\bar{p}_H|_T$ on any element $T \in \mathcal{T}_H$. It is easy to see that $\bar{p}_H$ is well defined and $\bar{p}_H \in M_H$. Then $p'_H = (p_H - \bar{p}_H) \in M^0_H$ defined in (3.129). Consequently, we can decompose p_H in a unique way as

$$p_H = \bar{p}_H + p'_H \text{ with } \bar{p}_H \in M_H \text{ and } p'_H \in M_H^0$$

By virtue of this decomposition, the term concerning p_H in (3.143) can be decomposed as

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} p_H \text{ div } \boldsymbol{v} = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} + \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} p'_H \text{ div } \boldsymbol{v} \qquad (3.144)$$

Now we compute the first term in the right hand side of (3.144). As $\bar{p}_H$ is constant on each $T \in \mathcal{T}_H$, we have

$$\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = \sum_{T\in\mathcal{T}_H} \bar{p}_H|_T \int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{v},\ \forall\boldsymbol{v} \in V^0_{H,bubble}$$

Taking advantage of Assumption 3.1.1, for any $\boldsymbol{v} \in V^0_{H,bubble}$, the divergence theorem and definition of $V^0_{H,bubble}$ reveals that

$$\int_{\partial(T\cap\Omega^\varepsilon)} \text{div } \boldsymbol{v} = \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v}\cdot\boldsymbol{n} = 0$$

Consequently, the first term in the hand side of (3.144) vanishes

$$\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = 0,\ \forall\boldsymbol{v} \in V^0_{H,bubble}$$

and then (3.143) reduces to: $\forall\, (\boldsymbol{v}, q) \in V^0_{H,bubble} \times M^0_H$

$$c^{St}_H((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p'_H \operatorname{div} \boldsymbol{v} - q \operatorname{div} \boldsymbol{u}_H\right) = 0 \tag{3.145}$$

In a second step, choosing an element $T \in \mathcal{T}_H$ and the test function $\boldsymbol{v} = \boldsymbol{0}$, for any $q \in M^0_H$ with q vanishing outside T, then (3.145) becomes

$$\int_{T\cap\Omega^\varepsilon} q \operatorname{div} \boldsymbol{u}_H = 0$$

We deduce from this equation that there exists a constant α^T in $T \cap \Omega^\varepsilon$ such that

$$\operatorname{div} \boldsymbol{u}_H = \alpha^T \text{ in } T \cap \Omega^\varepsilon \tag{3.146}$$

In a third step, we observe that for any face $E \in \mathcal{E}(T)$, there exist some (non unique) functions $\boldsymbol{v}_{E,i} \in \left(H^1(T\cap\Omega^\varepsilon)\right)^d$, $i = 1, \cdots, s$ such that

$$\begin{cases} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v}_{E,i} \cdot \boldsymbol{\omega}_{F,j} = \delta_{E,F}\delta_{i,j}, \ \forall F \in \mathcal{E}(T), \ \forall j = 1, \cdots, s \\ \int_{T\cap\Omega^\varepsilon} \boldsymbol{v}_{E,i} \cdot \boldsymbol{\varphi}_{T,l} = 0, \ \forall l = 1, \cdots, r \\ \boldsymbol{v}_{E,i} = \boldsymbol{0}, \text{ on } \partial B^\varepsilon \cap T \end{cases}$$

We observe that there exist some (non unique) functions $\boldsymbol{v}_{T,k} \in \left(H^1(T\cap\Omega^\varepsilon)\right)^d$, $k = 1, \cdots, r$ such that

$$\begin{cases} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v}_{T,k} \cdot \boldsymbol{\omega}_{F,j} = 0, \ \forall F \in \mathcal{E}(T), \ \forall j = 1, \cdots, s \\ \int_{T\cap\Omega^\varepsilon} \boldsymbol{v}_{T,k} \cdot \boldsymbol{\varphi}_{T,l} = \delta_{k,l}, \ \forall l = 1, \cdots, r \\ \boldsymbol{v}_{T,k} = \boldsymbol{0}, \text{ on } \partial B^\varepsilon \cap T \end{cases}$$

We denote by $V(T)$ the set of functions in $\left(H^1(T\cap\Omega^\varepsilon)\right)^d$ that vanish on $\partial B^\varepsilon \cap T$:

$$V(T) = \left\{\boldsymbol{v} : \left(H^1(T\cap\Omega^\varepsilon)\right)^d \text{ such that } \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T\right\}$$

It is easy to check that $V(T)$ can be decomposed as

$$V(T) = V_{\int 0}(T) \oplus \operatorname{span}\{\boldsymbol{v}_{E,i}, \ \boldsymbol{v}_{T,k}, \ \forall E \in \mathcal{E}(T), \ i = 1, \cdots, s, \ k = 1, \cdots, r\}$$

where

$$V_{\int 0}(T) = \left\{ \begin{array}{l} \boldsymbol{v} \in \left(H^1(T\cap\Omega^\varepsilon)\right)^d : \int_{E\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{E,i} = 0, \ \int_{T\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \varphi_{T,l} = 0, \ \boldsymbol{v} = \boldsymbol{0} \\ \text{on } \partial B^\varepsilon \cap T, \ \forall E \in \mathcal{E}(T), \ i = 1, \cdots, s, \ l = 1, \cdots, r \end{array} \right\}$$

By virtue of this decomposition, for any $\boldsymbol{v} \in V(T)$, there exist $\tilde{v} \in V_{\int 0}(T)$, $\beta_{F,1}, \cdots, \beta_{F,s} \in \mathbb{R}$ and $\beta_{T,1}, \cdots, \beta_{T,r} \in \mathbb{R}$ such that

$$\boldsymbol{v} = \tilde{\boldsymbol{v}} + \sum_{F\in\mathcal{E}(T)} \sum_{j=1}^{s} \beta_{F,j}\boldsymbol{v}_{F,j} + \sum_{k=1}^{r} \beta_{T,k}\boldsymbol{v}_{T,k} \tag{3.147}$$

Now we compute the coefficients $\beta_{E,i}$ for any $E \in \mathcal{E}(T)$ and $i = 1, \cdots, s$. Multiplying (3.147) by $\boldsymbol{\omega}_{E,i}$ and integrating over E yields:

$$\int_{E\cap\Omega^\varepsilon} \boldsymbol{v}\cdot\boldsymbol{\omega}_{E,i} = \int_{E\cap\Omega^\varepsilon} \tilde{\boldsymbol{v}}\cdot\boldsymbol{\omega}_{E,i} + \sum_{F\in\mathcal{E}(T)}\sum_{j=1}^{s}\beta_{F,j}\int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_{F,j}\cdot\boldsymbol{\omega}_{E,i} + \sum_{k=1}^{r}\beta_{T,k}\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}_{T,k}\cdot\boldsymbol{\omega}_{E,i}$$

Since $\tilde{\boldsymbol{v}} \in V^0_{H,bubble}$, the definition of $V^0_{H,bubble}$ implies that

$$\begin{cases} \int_{E\cap\Omega^\varepsilon} \tilde{\boldsymbol{v}}\cdot\boldsymbol{\omega}_{E,i} = 0 \\ \int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_{T,k}\cdot\boldsymbol{\omega}_{E,i} = 0 \\ \sum\limits_{F\in\mathcal{E}(T)}\sum\limits_{j=1}^{s}\beta_{F,j}\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}_{F,j}\cdot\boldsymbol{\omega}_{E,i} = \beta_{E,i} \end{cases} \Longrightarrow \int_{E\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{E,i} = \beta_{E,i} \tag{3.148}$$

Now we compute the coefficients $\beta_{T,l}$ for $l = 1, \cdots, r$. Multiplying (3.147) by $\boldsymbol{\varphi}_{T,l}$ and integrating over T yields:

$$\int_{T\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\varphi}_{T,l} = \int_{T\cap\Omega^\varepsilon}\tilde{\boldsymbol{v}}\cdot\boldsymbol{\varphi}_{T,l} + \sum_{F\in\mathcal{E}(T)}\sum_{j=1}^{s}\beta_{F,j}\int_{T\cap\Omega^\varepsilon}\boldsymbol{v}_{F,j}\cdot\boldsymbol{\varphi}_{T,l} + \sum_{j=1}^{s}\beta_{T,j}\int_{T\cap\Omega^\varepsilon}\boldsymbol{v}_{T,j}\cdot\boldsymbol{\varphi}_{T,l}$$

The fact that $\tilde{\boldsymbol{v}} \in V^0_{H,bubble}$ and the definition of $\boldsymbol{v}_{F,j}$ imply that

$$\begin{cases} \int_{T\cap\Omega^\varepsilon} \tilde{\boldsymbol{v}}\cdot\boldsymbol{\varphi}_{T,l} = 0 \\ \int_{T\cap\Omega^\varepsilon} \boldsymbol{v}_{F,j}\cdot\boldsymbol{\varphi}_{T,l} = 0 \\ \sum\limits_{j=1}^{s}\beta_{T,j}\int_{T\cap\Omega^\varepsilon}\boldsymbol{v}_{T,j}\cdot\boldsymbol{\varphi}_{T,l} = \beta_{T,l} \end{cases} \Longrightarrow \int_{T\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\varphi}_{T,l} = \beta_{T,l} \tag{3.149}$$

Let $\underline{\tilde{\boldsymbol{v}}}$ be the function defined on Ω^ε and equal to $\tilde{\boldsymbol{v}}$ on $T\cap\Omega^\varepsilon$ and $\mathbf{0}$ elsewhere. Hence $\underline{\tilde{\boldsymbol{v}}} \in V^0_{H,bubble}$. Taking $q = 0$, (3.145) implies that

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\underline{\tilde{\boldsymbol{v}}} - p'_H \text{ div } \underline{\tilde{\boldsymbol{v}}}\right) = \int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\tilde{\boldsymbol{v}} - p'_H \text{ div } \tilde{\boldsymbol{v}}\right) = 0$$

Substituting $\tilde{\boldsymbol{v}}$ defined in (3.147) into this equation, it is trivial to verify that

$$\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p'_H \text{ div } \boldsymbol{v}\right) = \sum_{F\in\mathcal{E}(T)}\sum_{j=1}^{s}\beta_{F,j}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{F,j} - p'_H \text{ div } \boldsymbol{v}_{F,j}\right) + \sum_{k=1}^{r}\beta_{T,k}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{T,k} - p'_H \text{ div } \boldsymbol{v}_{T,k}\right) \tag{3.150}$$

Denoting for any $F \in \mathcal{E}(T)$, $j = 1, \cdots, s$ and $k = 1, \cdots, r$

$$\lambda_{F,j} = \int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{F,j} - p'_H \text{ div } \boldsymbol{v}_{F,j}$$

$$\lambda_{T,k} = \int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{T,k} - p'_H \text{ div } \boldsymbol{v}_{T,k}$$

and substituting $\beta_{F,j}$ (3.148) and $\beta_{T,k}$ (3.149) into (3.150), we obtain

$$\int_{T\cap\Omega^\varepsilon} \left(\mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} - p'_H \text{ div } \boldsymbol{v}\right) = \sum_{F\in\mathcal{E}(T)} \sum_{j=1}^{s} \lambda_{F,j} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v}\cdot\boldsymbol{\omega}_{F,j} + \sum_{k=1}^{r} \lambda_{T,k} \int_{T\cap\Omega^\varepsilon} \boldsymbol{v}\cdot\boldsymbol{\varphi}_{T,k} \tag{3.151}$$

In a last step, in (3.151), taking $\boldsymbol{v} = \boldsymbol{0}$ on $\mathcal{E}(T)$ and integrating by parts the left hand side, we obtain

$$-\mu\Delta\boldsymbol{u}_H + \nabla p'_H = \sum_{k=1}^{r} \lambda_{T,k}\boldsymbol{\varphi}_{T,k} \text{ in } T\cap\Omega^\varepsilon \tag{3.152}$$

Then by writing the variational form of (3.152) for $\boldsymbol{v} \in V(T)$ and comparing with (3.151), it is easy to deduce that

$$\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - p'_H\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{E,1},\cdots,\boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E\in\mathcal{E}(T) \tag{3.153}$$

Finally, combining (3.146), (3.152) and (3.153), we obtain the following system

$$\begin{aligned} -\mu\Delta\boldsymbol{u}_H + \nabla p'_H &= \sum_{k=1}^{r} \lambda_{T,k}\boldsymbol{\varphi}_{T,k} \text{ in } T\cap\Omega^\varepsilon \\ \text{div } \boldsymbol{u}_H &= \alpha^T \text{ in } T\cap\Omega^\varepsilon \\ \boldsymbol{u}_H &= \boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T \\ \mu\nabla\boldsymbol{u}_H\boldsymbol{n} - p'_H\boldsymbol{n} &\in \text{span}\left\{\boldsymbol{\omega}_{E,1},\cdots,\boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E\in\mathcal{E}(T) \end{aligned}$$

On any element $T \in \mathcal{T}_H$, for any $\boldsymbol{u}_H \in V_H^{St+b}$ fixed in the formula above, now we check whether p'_H and $\boldsymbol{\lambda}_T$ are both uniquely determined by $\boldsymbol{u}_H$. Assuming that there exist $\left(\zeta_1^T,(\lambda_{T,k}^1)\right) \in M_H^0(T)\times\mathbb{R}^r$ and $\left(\zeta_2^T,(\lambda_{T,k}^2)\right) \in M_H^0(T)\times\mathbb{R}^r$ that verify:

$$\begin{aligned} \nabla\zeta_1^T &= \sum_{k=1}^{r} \lambda_{T,k}^1\boldsymbol{\varphi}_{T,k} + \mu\Delta\boldsymbol{u}_H \\ \nabla\zeta_2^T &= \sum_{k=1}^{r} \lambda_{T,k}^2\boldsymbol{\varphi}_{T,k} + \mu\Delta\boldsymbol{u}_H \end{aligned}$$

Subtracting these equations, we obtain

$$\nabla\left(\zeta_1^T - \zeta_2^T\right) = \sum_{k=1}^{r} \left(\lambda_{T,k}^1 - \lambda_{T,k}^2\right)\boldsymbol{\varphi}_{T,k}$$

The fact that $\sum_{k=1}^{r}\left(\lambda_{T,k}^1 - \lambda_{T,k}^2\right)\boldsymbol{\varphi}_{T,k} \in \mathbb{R}^d$ implies that $\left(\zeta_1^T - \zeta_2^T\right) \in \mathbb{P}_1(T)$. Meanwhile, $\zeta_1^T \in M_H^0(T)$ and $\zeta_2^T \in M_H^0(T)$. Combining these two arguments, we can not conclude that $\zeta_1^T - \zeta_2^T = 0$ and $(\lambda_{T,k}^1) = (\lambda_{T,k}^2)$. Thus the uniqueness of p'_H and $\boldsymbol{\lambda}_T$ is not guaranteed and the linear operator π_H defined by Definition 3.5.3 is not well defined. Consequently, we are not able to prove that $(\boldsymbol{u}_H, p_H) \in X_H^{St+b}$ in the sense of definition (3.130) belongs to the space defined by (3.133).

It is easy to notice that $\zeta_1^T - \zeta_2^T = 0$ if and only if spaces $M_H^0(T)$ and $\mathbb{P}_1(T)$ are orthogonal. This fact implies that the definition of X_H^{St+b} and $X_{H,bubble}^0$ need to be revised. We will redefine these spaces in section 3.6, leading to a more innovative and more general multiscale method which can effectively improve the accuracy of numerical solutions.

3.5.3 The local problems defined by Stokes equations

Inspired by bubble functions defined in [105, 113], we have proposed to construct a basis of V_H^{St+b} by solving the following local problems.

The strong form We recall the multiscale basis functions $\mathbf{\Phi}_{E,i}^{St}$ which were already defined by (3.39): for any $E \in \mathcal{E}_H$, for $i = 1, \cdots, s$, we construct $\mathbf{\Phi}_{E,i}^{St} : \Omega^\varepsilon \to \mathbb{R}^d$ and the accompanying pressure $\pi_{E,i}^{St} : \Omega^\varepsilon \to \mathbb{R}$ such that $\mathbf{\Phi}_{E,i}^{St}$ and $\pi_{E,i}^{St}$ vanish outside the coarse element $T_k \subset \omega_E$ for $k \in \{1, 2\}$ (only one coarse element if $E \in \partial\Omega$) and solve on T_k:

$$\begin{cases} -\mu\Delta\mathbf{\Phi}_{E,i}^{St} + \nabla\pi_{E,i}^{St} = \mathbf{0} \text{ in } T_k \cap \Omega^\varepsilon, \\ \text{div } \mathbf{\Phi}_{E,i}^{St} = \alpha_{E,i} \text{ in } T_k \cap \Omega^\varepsilon, \\ \mu\nabla\mathbf{\Phi}_{E,i}^{St}\boldsymbol{n} - \pi_{E,i}^{St}\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\right\} \text{ on } F \cap \Omega^\varepsilon, \ \forall F \in \mathcal{E}(T_k), \\ \mathbf{\Phi}_{E,i}^{St} = \mathbf{0} \text{ on } \partial B^\varepsilon \cap T_k, \\ \int_{F\cap\Omega^\varepsilon} \mathbf{\Phi}_{E,i}^{St} \cdot \boldsymbol{\omega}_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases} \quad \forall F \in \mathcal{E}\left(T_k\right), \ j = 1, \cdots, s. \\ \int_{T_k\cap\Omega^\varepsilon} \pi_{E,i}^{St} = 0. \end{cases} \tag{3.154}$$

where $\mathcal{E}(T_k)$ is the set of faces composing ∂T_k. The constant $\alpha_{E,i}$ depends on T_k and satisfies $\int_{T_k\cap\Omega^\varepsilon} \alpha_{E,i} = \int_{\partial(T_k\cap\Omega^\varepsilon)} \mathbf{\Phi}_{E,i}^{St} \cdot \boldsymbol{n}$.

Now we construct bubble functions associated to elements of the coarse mesh. For each $T \in \mathcal{T}_H$, for $k = 1, \cdots, r$, the support of $\mathbf{\Psi}_{T,k}$ is reduced to $T \cap \Omega^\varepsilon$. We construct $\mathbf{\Psi}_{T,k} : \Omega^\varepsilon \to \mathbb{R}^d$ and $\pi_{T,k} : \Omega^\varepsilon \to \mathbb{R}$ as solutions of

$$\begin{cases} -\mu\Delta\mathbf{\Psi}_{T,k} + \nabla\pi_{T,k} = \boldsymbol{e}_k \text{ in } T \cap \Omega^\varepsilon, \\ \text{div } \mathbf{\Psi}_{T,k} = \alpha_{T,k} \text{ in } T \cap \Omega^\varepsilon, \\ \mu\nabla\mathbf{\Psi}_{T,k}\boldsymbol{n} - \pi_{T,k}\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\right\} \text{ on } F \cap \Omega^\varepsilon, \ \forall F \in \mathcal{E}(T), \\ \mathbf{\Psi}_{T,k} = \mathbf{0} \text{ on } \partial B^\varepsilon \cap T, \\ \int_{F\cap\Omega^\varepsilon} \mathbf{\Psi}_{T,k} = \mathbf{0} \ \forall F \in \mathcal{E}\left(T\right), \\ \int_{T\cap\Omega^\varepsilon} \pi_{T,k} = 0. \end{cases} \tag{3.155}$$

where $\mathcal{E}(T)$ is the set of faces composing ∂T and the constant $\alpha_{T,k}$ satisfies $\int_{T\cap\Omega^\varepsilon} \alpha_{T,k} = \int_{\partial(T\cap\Omega^\varepsilon)} \mathbf{\Psi}_{T,k} \cdot \boldsymbol{n}$.

The weak form The weak form of system (3.154) reads: for any face $E \in \mathcal{E}_H$, for $i = 1, \cdots, s$, on the coarse element $T_k \subset \omega_E$ for $k \in \{1, 2\}$ (only one coarse element if $E \in \partial\Omega$), find $\mathbf{\Phi}_{E,i}^{St} \in \left(H^1\left(T_k \cap \Omega^\varepsilon\right)\right)^d$ such that $\mathbf{\Phi}_{E,i}^{St} = \mathbf{0}$ on $T_k \cap B^\varepsilon$, $\pi_{E,i}^{St} \in L_0^2\left(T \cap \Omega^\varepsilon\right)$ and Lagrange multipliers $\lambda_{F,1},\cdots,\lambda_{F,s}\in\mathbb{R}$ for $F\in\mathcal{E}(T_k)$ such that

$$\begin{aligned}\int_{T_k\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{\Phi}^{St}_{E,i}:\nabla\boldsymbol{v}-\int_{T_k\cap\Omega^\varepsilon}\pi^{St}_{E,i}\operatorname{div}\boldsymbol{v}+\sum_{F\in\mathcal{E}(T_k)}\sum_{j=1}^{s}\lambda_{F,j}\int_{F\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{F,j}&=0,\\ \int_{T_k\cap\Omega^\varepsilon}q\operatorname{div}\boldsymbol{\Phi}^{St}_{E,i}&=0,\\ \sum_{F\in\mathcal{E}(T_k)}\sum_{j=1}^{s}\mu_{F,j}\int_{F\cap\Omega^\varepsilon}\boldsymbol{\Phi}^{St}_{E,i}\cdot\boldsymbol{\omega}_{F,j}&=\mu_{E,i}\end{aligned}\tag{3.156}$$

for all $\boldsymbol{v}\in\left(H^1\left(T_k\cap\Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{v}=\boldsymbol{0}$ on $\partial B^\varepsilon\cap T_k$, $q\in M^0_H\left(T_k\right)$, $\mu_{F,j}\in\mathbb{R}$ for all $F\in\mathcal{E}\left(T_k\right)$ and $j=1,\cdots,s$.

The weak form of system (3.155) is: for each $T\in\mathcal{T}_H$, for $k=1,\cdots,r$, find $\boldsymbol{\Psi}_{T,k}\in\left(H^1\left(T\cap\Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{\Psi}_{T,k}=\boldsymbol{0}$ on $T\cap B^\varepsilon$, $\pi_{T,k}\in L^2_0\left(T\cap\Omega^\varepsilon\right)$ and Lagrange multipliers $\lambda_{F,1},\cdots,\lambda_{F,s}\in\mathbb{R}$ for $F\in\mathcal{E}(T)$ such that

$$\begin{aligned}\int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{\Psi}_{T,k}:\nabla\boldsymbol{v}-\int_{T\cap\Omega^\varepsilon}\pi_{T,k}\operatorname{div}\boldsymbol{v}+\sum_{F\in\mathcal{E}(T)}\lambda_{F,j}\int_{F\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{F,j}&=\int_{T\cap\Omega^\varepsilon}\boldsymbol{e}_k\cdot\boldsymbol{v},\\ \int_{T\cap\Omega^\varepsilon}q\operatorname{div}\boldsymbol{\Psi}_{T,k}&=0,\\ \sum_{F\in\mathcal{E}(T)}\boldsymbol{\mu}_F\cdot\int_{F\cap\Omega^\varepsilon}\boldsymbol{\Psi}_{T,k}&=0\end{aligned}$$

for all $\boldsymbol{v}\in\left(H^1\left(T\cap\Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{v}=\boldsymbol{0}$ on $\partial B^\varepsilon\cap T$, $q\in M^0_H(T)$, $\boldsymbol{\mu}_F\in\mathbb{R}^d$ for all $F\in\mathcal{E}(T)$.

3.5.4 The contribution of bubble functions

Now we try to find out the relation between basis functions $\boldsymbol{\Phi}^{St}_{E,i}$ for $E\in\mathcal{E}_H$ and $i=1,\cdots,s$ defined by (3.154) and $\boldsymbol{\Psi}_{T,k}$ for $T\in\mathcal{T}_H$ and $k=1,\cdots,r$ defined by (3.155). In (3.156), taking the test function $\boldsymbol{v}=\boldsymbol{\Psi}_{T,k}$ for $k=1,\cdots,r$, then (3.156) becomes:

$$\int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{\Phi}^{St}_{E,i}:\nabla\boldsymbol{\Psi}_{T,k}-\int_{T\cap\Omega^\varepsilon}\pi^{St}_{E,i}\operatorname{div}\boldsymbol{\Psi}_{T,k}+\sum_{F\in\mathcal{E}(T)}\sum_{t=1}^{s}\lambda_{F,t}\int_{F\cap\Omega^\varepsilon}\boldsymbol{\Psi}_{T,k}\cdot\boldsymbol{\omega}_{F,t}=0\tag{3.157}$$

System (3.155) reveals that div $\boldsymbol{\Psi}_{T,k}$ is constant on T and $\int_{T\cap\Omega^\varepsilon}\pi^{St}_{E,i}=0$, we can thus deduce that

$$\int_{T\cap\Omega^\varepsilon}\pi^{St}_{E,i}\operatorname{div}\boldsymbol{\Psi}_{T,k}=0\tag{3.158}$$

Making use of Assumption 3.5.1, system (3.155) implies that

$$\sum_{F\in\mathcal{E}(T)}\sum_{t=1}^{s}\lambda_{F,t}\int_{F\cap\Omega^\varepsilon}\boldsymbol{\Psi}_{T,k}\cdot\boldsymbol{\omega}_{F,t}=0\tag{3.159}$$

Combining (3.158) and (3.159), equation (3.157) reduces to

$$\int_{T\cap\Omega^\varepsilon} \mu \nabla \mathbf{\Phi}_{E,i}^{St} : \nabla \mathbf{\Psi}_{T,k} = 0 \tag{3.160}$$

Surprisingly, this equation reveals that $\mathbf{\Phi}_{E,i}^{St}$ and $\mathbf{\Psi}_{T,k}$ are "orthogonal" with respect to the bilinear form $\int_{T\cap\Omega^\varepsilon} \nabla \boldsymbol{u} : \nabla \boldsymbol{v}$ for $\boldsymbol{u} = \mathbf{\Phi}_{E,i}^{St}$ and $\boldsymbol{v} = \mathbf{\Psi}_{T,k}$. Curiously, numerical experiments (see Chapter 6) show that there are no significant differences between numerical results given by using V_H^{St} and V_H^{St+b} when solving a Stokes problem on the coarse mesh.

Making use of basis functions defined by (3.154)–(3.155), we reconstruct the fine-scale velocity on each coarse element $T \in \mathcal{T}_H$ by

$$\boldsymbol{u}_H|_T = \sum_{E\in\mathcal{E}(T)} \sum_{i=1}^{s} u_{E,i} \mathbf{\Phi}_{E,i}^{St} + \sum_{k=1}^{r} u_{T,k} \mathbf{\Psi}_{T,k}$$

Then we found that the contribution of the second term is negligible compared to that of the first term in the right hand side of the equation above. Overall, our numerical results show that the addition of bubble functions defined by (3.155) does not improve the accuracy of velocity and pressure.

3.6 The high-order Crouzeix-Raviart multiscale finite element method defined by Stokes equations

Until now, we have presented two methods to enrich the approximation space of velocity V_H whereas the approximation space of pressure M_H still remains piecewise constant on the coarse mesh. In order to further improve the accuracy of the Crouzeix-Raviart multiscale finite element methods, we propose an innovative high-order multiscale method where both the approximation spaces of velocity and pressure are enriched. As far as we know, it is the first time that this multiscale method is proposed for flow problems.

We mention that [50] proposes recently a multiscale hydrid high-order method for highly oscillatory elliptic problems. This method can be considered as a first attempt of the generalization of the Crouzeix–Raviart MsFEM to arbitrary orders of approximation. This method share some similarities with our high-order method in the use of weighting functions defined by polynomials of higher degrees. But the method in [50] is defined only for elliptic problems in the framework of a hybrid high-order method, whereas we develop in our work a higher-order method for flow problems in the framework of MsFEM.

3.6.1 The construction of the approximation space $\widehat{X}_H^{St}$

We first introduce some notations for later use. For any integer n and any integer $1 \leq l \leq d$, we denote by $\mathbb{P}_n^l$ the linear space spanned by l-variate polynomial functions of degree at most n. The dimension of $\mathbb{P}_n^l$ is

$$N_n^l := \dim\left(\mathbb{P}_n^l\right) = \binom{n+l}{n}$$

For any $T \in \mathcal{T}_H$, we denote by $\mathbb{P}_n^d(T)$ the restriction to T of polynomials in $\mathbb{P}_n^l$. For any $F \in \mathcal{E}_H$, we denote by $\mathbb{P}_n^{d-1}(F)$ the restriction to F of polynomials of $\mathbb{P}_n^l$. For the sake of simplicity, we denote $\mathbb{P}_n^d(T)$ and $\mathbb{P}_n^{d-1}(F)$ respectively by $\mathbb{P}_n(T)$ and $\mathbb{P}_n(F)$.

We now define weighting functions for the velocity and pressure in what follows. Let s be a positive integer and d be the dimension of heterogeneous media. We associate the vector-valued function $\boldsymbol{\omega}_{E,i}$: $E \to \mathbb{R}^d$ to each face $E \in \mathcal{E}_H$ and $i = 1, \cdots, s$. Let r be a positive integer. We associate the vector-valued functions $\boldsymbol{\varphi}_{T,k} : T \to \mathbb{R}^d$ to each coarse element $T \in \mathcal{T}_H$ and $k = 1, \cdots, r$. Let t be a positive integer. We associate the scalar functions $\varpi_{T,j}$: $T \to \mathbb{R}$ to each coarse element $T \in \mathcal{T}_H$ and $j = 1, \cdots, t$.

Assumption 3.6.1. *For $n = 0$, for any $T \in \mathcal{T}_H$ and for any $E \in \mathcal{E}_H$, we choose*

$$\begin{cases} s = d : \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1, \cdots, \boldsymbol{\omega}_{E,d} = \boldsymbol{e}_d. \\ r = 1 : \ \boldsymbol{\varphi}_{T,1} = 0 \\ t = 1 : \ \varpi_{T,1} = 1 \end{cases} \tag{3.161}$$

We notice that the choice of weighting functions above is the same as that of the case $s = 2$ or $s = 3$ defined in (3.7) and (3.9).

Assumption 3.6.2. *For $n \geq 1$, for any $E \in \mathcal{E}_H$ and for any $T \in \mathcal{T}_H$, we assume that*

1. *$s = d \cdot N_n^{d-1}$ and $(\boldsymbol{\omega}_{E,i})_{1 \leq i \leq s}$ is a set of basis functions of the space $(\mathbb{P}_n(E))^d$.*
2. *$r = d \cdot N_{n-1}^d$ and $(\boldsymbol{\varphi}_{T,k})_{1 \leq k \leq r}$ is a set of basis functions of the space $(\mathbb{P}_{n-1}(T))^d$.*
3. *$t = N_n^d$ and $(\varpi_{T,j})_{1 \leq j \leq t}$ is a set of basis functions of the space $\mathbb{P}_n(T)$.*

Definition 3.6.1. *We introduce the extended velocity space $\widehat{V}_H^{ext}$:*

$$\widehat{V}_H^{ext} = \left\{ \begin{array}{l} \boldsymbol{u} \in \left(L^2(\Omega^\varepsilon)\right)^d \text{ such that } \boldsymbol{u}\,|_T \in \left(H^1(T \cap \Omega^\varepsilon)\right)^d \text{ for any } T \in \mathcal{T}_H, \\ \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial B^\varepsilon, \ \int_{E \cap \Omega^\varepsilon} [[\boldsymbol{u}]] \cdot \boldsymbol{\omega}_{E,j} = 0 \text{ for all } E \in \mathcal{E}_H, \ j = 1, \cdots, s. \end{array} \right\}$$

where $[[\boldsymbol{u}]]$ denotes the "jump" of $\boldsymbol{u}$ across an internal face and $[[\boldsymbol{u}]] = \boldsymbol{u}$ on the boundary $\partial\Omega$.

We recall that the pressure space M is defined by

$$M = L_0^2(\Omega^\varepsilon) = \left\{ p \in L^2(\Omega^\varepsilon) \text{ such that } \int_{\Omega^\varepsilon} p = 0 \right\}$$

Then the extended velocity-pressure space X_H^{ext} can be written as

$$X_H^{ext} = \widehat{V}_H^{ext} \times M$$

We want to decompose X_H^{ext} into the sum of a finite dimensional subspace $\widehat{X}_H$ of coarse scales and an infinite dimensional subspace $\widehat{X}_H^0$ of unsolved fine scales:

$$X_H^{ext} = \widehat{X}_H \oplus \widehat{X}_H^0$$

Taking into account Assumption 3.6.1 and Assumption 3.6.2, we introduce the fine scale subspace $\widehat{X}_H^0 = \widehat{V}_H^0 \times \widehat{M}_H^0$ with

$$\widehat{V}_H^0 = \left\{ \begin{array}{l} \boldsymbol{u} \in \widehat{V}_H^{ext} \text{ such that } \int_{E \cap \Omega^\varepsilon} \boldsymbol{u} \cdot \boldsymbol{\omega}_{E,j} = 0, \ \int_{T \cap \Omega^\varepsilon} \boldsymbol{u} \cdot \boldsymbol{\varphi}_{T,k} = 0, \\ \forall T \in \mathcal{T}_H, \ \forall E \in \mathcal{E}_H, \ j = 1, \cdots, s, \ k = 1, \cdots, r. \end{array} \right\} \tag{3.162}$$

$$\widehat{M}_H^0 = \left\{ p \in M \text{ such that } \int_{T \cap \Omega^\varepsilon} p \varpi_{T,i} = 0, \ \forall T \in \mathcal{T}_H, \ i = 1, \cdots, t. \right\} \tag{3.163}$$

Besides, we define $\widehat{M}_H^0(T)$ by

$$\widehat{M}_H^0(T) = \{ \ p \in L^2(T \cap \Omega^\varepsilon) \text{ such that } \int_{T \cap \Omega^\varepsilon} p \varpi_{T,i} = 0, \ \forall T \in \mathcal{T}_H, \ i = 1, \cdots, t. \ \} \tag{3.164}$$

Since this high-order Crouzeix-Raviart multiscale finite element method is constructed by Stokes equations, we denote the space $\widehat{X}_H$ by $\widehat{X}_H^{St}$.

Definition 3.6.2. *The subspace $\widehat{X}_H^{St}$ is defined as the "orthogonal" complement of $\widehat{X}_H^0$ with respect to the bilinear form $c_H^{St}(.,.)$:*

$$(\boldsymbol{u}_H, p_H) \in \widehat{X}_H^{St} \Longleftrightarrow c_H^{St}\left((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)\right) = 0,\ \forall (\boldsymbol{v}, q) \in \widehat{X}_H^0 \tag{3.165}$$

where $c_H^{St}(.,.)$ is defined by

$$c_H^{St}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} (\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p_H\, div\, \boldsymbol{v} - q\, div\, \boldsymbol{u}_H)$$

Definition 3.6.3. *Let the functional spaces $\widehat{M}_H \subset M$ and $\widehat{V}_H^{St} \subset \widehat{V}_H^{ext}$ be defined by*

$$\widehat{M}_H = \left\{q \in L_0^2(\Omega)\ \text{such that}\ q\mid_T \in \mathbb{P}_n(T),\ \forall T \in \mathcal{T}_H\right\} \tag{3.166}$$

$$\widehat{V}_H^{St} = \left\{\begin{array}{l} \boldsymbol{v} \in \left(L^2(\Omega^\varepsilon)\right)^d : \forall T \in \mathcal{T}_H,\ \exists \zeta^T \in \widehat{M}_H^0(T)\ \text{such that} \\ -\mu\Delta\boldsymbol{v} + \nabla\zeta^T \in \operatorname{span}\{\boldsymbol{\varphi}_{T,1}, \cdots, \boldsymbol{\varphi}_{T,r}\}\ \text{in}\ T\cap\Omega^\varepsilon \\ div\,\boldsymbol{v} \in \operatorname{span}\{\varpi_{T,1}, \cdots, \varpi_{T,t}\}\ \text{in}\ T\cap\Omega^\varepsilon \\ \boldsymbol{v} = \boldsymbol{0}\ \text{on}\ \partial B^\varepsilon \cap T \\ \mu\nabla\boldsymbol{v}\boldsymbol{n} - \zeta^T\boldsymbol{n} \in \operatorname{span}\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\}\ \text{on}\ E\cap\Omega^\varepsilon\ \forall E \in \mathcal{E}(T) \end{array}\right\} \tag{3.167}$$

where $\mathcal{E}(T)$ is the set of faces composing ∂T.

Definition 3.6.4. *For any $\boldsymbol{v} \in \widehat{V}_H^{St}$, on any element $T \in \mathcal{T}_H$, definition (3.167) implies that there exists a function ζ^T which is uniquely determined by $\boldsymbol{v}$. We can thus define a linear operator $\pi_H : \widehat{V}_H^{St} \to \widehat{M}_H^0$ such that for any $\boldsymbol{v} \in \widehat{V}_H^{St}$, $\pi_H(\boldsymbol{v}) = \zeta^T$ on any element $T \in \mathcal{T}_H$.*

Theorem 3.6.1. *Using $\widehat{M}_H$ (3.166) and $\widehat{V}_H^{St}$ (3.167), we define $\underline{\widehat{X}}_H^{St}$ by*

$$\underline{\widehat{X}}_H^{St} = \operatorname{span}\left\{(\boldsymbol{u}_H, \pi_H(\boldsymbol{u}_H) + \bar{p}_H), \boldsymbol{u}_H \in \widehat{V}_H^{St},\ \bar{p}_H \in \widehat{M}_H\right\} \tag{3.168}$$

then the space $\widehat{X}_H^{St}$ defined in (3.165) satisfies the following property

$$\widehat{X}_H^{St} = \underline{\widehat{X}}_H^{St} \tag{3.169}$$

This theorem is new and now we provide its detailed proof in what follows.

Proof. We first prove that $(\boldsymbol{u}_H, p_H) \in \widehat{X}_H^{St}$ in the sense of definition (3.165) belongs to the space defined by (3.168). Let $(\boldsymbol{u}_H, p_H) \in \widehat{X}_H^{St}$ in the sense of definition (3.165), i.e. $\forall (\boldsymbol{v}, q) \in \widehat{X}_H^0$:

$$c_H^{St}((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)) = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} (\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p_H \operatorname{div} \boldsymbol{v} - q \operatorname{div} \boldsymbol{u}_H) = 0 \tag{3.170}$$

Choosing $\bar{p}_H \in \widehat{M}_H$, the fact that $\widehat{M}_H$ and $\widehat{M}_H^0$ are orthogonal implies that $p'_H = (p_H - \bar{p}_H)$ is well-defined and $p'_H \in \widehat{M}_H^0$ defined in (3.163). Thus we can decompose the pressure p_H in a unique way as

$$p_H = \bar{p}_H + p'_H \text{ with } \bar{p}_H \in \widehat{M}_H \text{ and } p'_H \in \widehat{M}_H^0$$

By virtue of this decomposition, the term concerning p_H in (3.170) can be decomposed as

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} p_H \text{ div } \boldsymbol{v} = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} + \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} p'_H \text{ div } \boldsymbol{v} \tag{3.171}$$

Now we compute the first term in the right hand side of (3.171). Integrating by parts yields:

$$\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v}\cdot\boldsymbol{n}\bar{p}_H - \int_{T\cap\Omega^\varepsilon} \boldsymbol{v}\cdot \text{ grad } \bar{p}_H,\ \forall \boldsymbol{v}\in \widehat{V}_H^0 \tag{3.172}$$

Since $\bar{p}_H \in \widehat{M}_H$, definition of $\widehat{M}_H$ (3.166) implies that for any $T\in\mathcal{T}_H$, $\bar{p}_H|_T \in \mathbb{P}_n(T)$. Thus it is easy to see that for any $E\in\mathcal{E}(T)$, $\bar{p}_H|_E\in\mathbb{P}_n(E)$. Making use of Assumption 3.1.1 and the item 1 of Assumption 3.6.2, we have $\boldsymbol{n}\bar{p}_H|_E \in \text{span}\{\boldsymbol{\omega}_{E,1},\cdots,\boldsymbol{\omega}_{E,s}\}$. Then definition of $\widehat{V}_H^0$ implies that

$$\int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v}\cdot\boldsymbol{n}\bar{p}_H = 0$$

Then (3.172) reduces to

$$\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = -\int_{T\cap\Omega^\varepsilon} \boldsymbol{v}\cdot \text{ grad } \bar{p}_H$$

As $\bar{p}_H|_T \in \mathbb{P}_n(T)$, it is obvious that grad $\bar{p}_H \in \mathbb{P}_{n-1}(T)$ on $T\in\mathcal{T}_H$. Making use of the item 2 of Assumption 3.6.2, definition of $\widehat{V}_H^0$ implies that

$$\int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{ div } \boldsymbol{v} = -\int_{T\cap\Omega^\varepsilon} \boldsymbol{v}\cdot \text{ grad } \bar{p}_H = 0$$

As a result, (3.171) equals to

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} p_H \text{ div } \boldsymbol{v} = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} p'_H \text{ div } \boldsymbol{v}$$

and (3.170) reduces to: $\forall(\boldsymbol{v},q)\in\widehat{X}_H^0$,

$$c_H^{St}\left((\boldsymbol{u}_H,p_H),(\boldsymbol{v},q)\right) = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v} - p'_H \text{ div } \boldsymbol{v} - q \text{ div } \boldsymbol{u}_H\right) = 0 \tag{3.173}$$

In a second step, choosing an element $T\in\mathcal{T}_H$ and the test function $\boldsymbol{v}=\boldsymbol{0}$, for any $q\in\widehat{M}_H^0$ with q vanishing outside T, (3.173) becomes

$$\int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0$$

Then by making use of the item 3 of Assumption 3.6.2, it is straightforward to verify that

$$\text{div } \boldsymbol{u}_H \in \mathbb{P}_n(T) \text{ in } T\cap\Omega^\varepsilon,\ \text{i.e. div } \boldsymbol{u}_H \in \text{span}\{\varpi_{T,1},\cdots,\varpi_{T,t}\} \tag{3.174}$$

In a third step, we observe that for any face $E \in \mathcal{E}(T)$, there exist some (non unique) functions $\boldsymbol{v}_{E,i} \in \left(H^1\left(T \cap \Omega^\varepsilon\right)\right)^d$, $i = 1, \cdots, s$ such that

$$\begin{cases} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v}_{E,i} \cdot \boldsymbol{\omega}_{F,j} = \delta_{E,F}\delta_{i,j}, \ \forall F \in \mathcal{E}(T), \ \forall j = 1, \cdots, s \\ \int_{T\cap\Omega^\varepsilon} \boldsymbol{v}_{E,i} \cdot \boldsymbol{\varphi}_{T,l} = 0, \ \forall l = 1, \cdots, r \\ \boldsymbol{v}_{E,i} = \boldsymbol{0}, \ \text{on } \partial B^\varepsilon \cap T \end{cases}$$

We observe that there exist some (non unique) functions $\boldsymbol{v}_{T,k} \in \left(H^1\left(T \cap \Omega^\varepsilon\right)\right)^d$, $k = 1, \cdots, r$ such that

$$\begin{cases} \int_{F\cap\Omega^\varepsilon} \boldsymbol{v}_{T,k} \cdot \boldsymbol{\omega}_{F,j} = 0, \ \forall F \in \mathcal{E}(T), \ \forall j = 1, \cdots, s \\ \int_{T\cap\Omega^\varepsilon} \boldsymbol{v}_{T,k} \cdot \boldsymbol{\varphi}_{T,l} = \delta_{k,l}, \ \forall l = 1, \cdots, r \\ \boldsymbol{v}_{T,k} = \boldsymbol{0}, \ \text{on } \partial B^\varepsilon \cap T \end{cases}$$

We denote by $V(T)$ the set of functions in $\left(H^1\left(T \cap \Omega^\varepsilon\right)\right)^d$ vanishing on $\partial B^\varepsilon \cap T$:

$$V(T) = \left\{ \boldsymbol{v} \in \left(H^1(T \cap \Omega^\varepsilon)\right)^d \text{ such that } \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T \right\}$$

It is easy to check that the space $V(T)$ can be decomposed as

$$V(T) = V_{\int 0}(T) \oplus \operatorname{span}\left\{\boldsymbol{v}_{E,i}, \ \boldsymbol{v}_{T,k}, \ \forall E \in \mathcal{E}(T), \ i = 1, \cdots, s, \ k = 1, \cdots, r\right\}$$

where

$$V_{\int 0}(T) = \left\{ \begin{array}{l} \boldsymbol{v} \in \left(H^1(T \cap \Omega^\varepsilon)\right)^d : \int_{E\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{E,i} = 0, \ \int_{T\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\varphi}_{T,l} = 0, \ \boldsymbol{v} = \boldsymbol{0} \\ \text{on } \partial B^\varepsilon \cap T, \ \forall E \in \mathcal{E}(T), \ i = 1, \cdots, s, \ l = 1, \cdots, r. \end{array} \right\}$$

By virtue of this decomposition, for any $\boldsymbol{v} \in V(T)$, there exist $\tilde{\boldsymbol{v}} \in V_{\int 0}(T)$, $\beta_{F,1}, \cdots, \beta_{F,s} \in \mathbb{R}$ for $F \in \mathcal{E}_H$ and $\beta_{T,1}, \cdots, \beta_{T,r} \in \mathbb{R}$ such that

$$\boldsymbol{v} = \tilde{\boldsymbol{v}} + \sum_{F\in\mathcal{E}_H} \sum_{j=1}^{s} \beta_{F,j} \boldsymbol{v}_{F,j} + \sum_{k=1}^{r} \beta_{T,k} \boldsymbol{v}_{T,k} \tag{3.175}$$

Now we compute $\beta_{E,i}$ for any $E \in \mathcal{E}(T)$ and $i = 1, \cdots, s$. Multiplying (3.175) by $\boldsymbol{\omega}_{E,i}$ and integrating over E yields

$$\begin{aligned} \int_{E\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{E,i} = \int_{E\cap\Omega^\varepsilon} \tilde{\boldsymbol{v}} \cdot \boldsymbol{\omega}_{E,i} &+ \sum_{F\in\mathcal{E}_H} \sum_{j=1}^{s} \beta_{F,j} \int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_{F,j} \cdot \boldsymbol{\omega}_{E,i} \\ &+ \sum_{k=1}^{r} \beta_{T,k} \int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_{T,k} \cdot \boldsymbol{\omega}_{E,i} \end{aligned}$$

Making use of definition of $\widehat{V}_H^0$ (3.162), it is trivial to verify that

$$\begin{cases} \int_{E\cap\Omega^\varepsilon} \tilde{\boldsymbol{v}} \cdot \boldsymbol{\omega}_{E,i} = 0 \text{ for } \tilde{\boldsymbol{v}} \in V_{\int 0}(T) \\ \sum\limits_{k=1}^{r} \beta_{T,k} \int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_{T,k} \cdot \boldsymbol{\omega}_{E,i} = 0 \\ \sum\limits_{F\in\mathcal{E}_H} \sum\limits_{j=1}^{s} \beta_{F,j} \int_{E\cap\Omega^\varepsilon} \boldsymbol{v}_{F,j} \cdot \boldsymbol{\omega}_{E,i} = \beta_{E,i} \end{cases} \Longrightarrow \int_{E\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \boldsymbol{\omega}_{E,i} = \beta_{E,i} \tag{3.176}$$

Then we compute $\beta_{T,l}$ for any $l = 1, \cdots, r$. Multiplying (3.175) by $\boldsymbol{\varphi}_{T,l}$ and integrating over $T \in \mathcal{T}_H$ yields

$$\int_{T\cap\Omega^\varepsilon} \boldsymbol{v}\cdot\boldsymbol{\varphi}_{T,l} = \int_{T\cap\Omega^\varepsilon} \tilde{\boldsymbol{v}}\cdot\boldsymbol{\varphi}_{T,l} + \sum_{F\in\mathcal{E}_H}\sum_{j=1}^{s}\beta_{F,j}\int_{T\cap\Omega^\varepsilon} \boldsymbol{v}_{F,j}\cdot\boldsymbol{\varphi}_{T,l} + \sum_{k=1}^{r}\beta_{T,k}\int_{T\cap\Omega^\varepsilon}\boldsymbol{v}_{T,k}\cdot\boldsymbol{\varphi}_{T,l}$$

Making use of definition of $\widehat{V}_H^0$ (3.162), it is straightforward to check that

$$\begin{cases} \int_{T\cap\Omega^\varepsilon} \tilde{\boldsymbol{v}}\cdot\boldsymbol{\varphi}_{T,l} = 0 \text{ for } \tilde{\boldsymbol{v}} \in V_{\int 0}(T) \\ \sum\limits_{F\in\mathcal{E}_H}\sum\limits_{j=1}^{s}\beta_{F,j}\int_{T\cap\Omega^\varepsilon}\boldsymbol{v}_{F,j}\cdot\boldsymbol{\varphi}_{T,l} = 0 \\ \sum\limits_{k=1}^{r}\beta_{T,k}\int_{T\cap\Omega^\varepsilon}\boldsymbol{v}_{T,k}\cdot\boldsymbol{\varphi}_{T,l} = \beta_{T,l} \end{cases} \Longrightarrow \int_{T\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\varphi}_{T,l} = \beta_{T,l} \tag{3.177}$$

Now let $\underline{\tilde{\boldsymbol{v}}}$ be the function which equals to $\tilde{\boldsymbol{v}}$ on $T\cap\Omega^\varepsilon$ and $\mathbf{0}$ elsewhere. Hence it is obvious that $\underline{\tilde{\boldsymbol{v}}} \in \widehat{V}_H^0$. Taking $q = 0$, (3.173) becomes

$$\int_{\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\underline{\tilde{\boldsymbol{v}}} - p'_H \operatorname{div}\underline{\tilde{\boldsymbol{v}}}\right) = \int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\tilde{\boldsymbol{v}} - p'_H \operatorname{div}\tilde{\boldsymbol{v}}\right) = 0 \tag{3.178}$$

Substituting $\tilde{\boldsymbol{v}}$ defined in (3.175) into (3.178), we obtain

$$\begin{aligned} \int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p'_H \operatorname{div}\boldsymbol{v}\right) = & \sum_{F\in\mathcal{E}(T)}\sum_{j=1}^{s}\beta_{F,j}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{F,j} - p'_H \operatorname{div}\boldsymbol{v}_{F,j}\right) \\ & + \sum_{k=1}^{r}\beta_{T,k}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{T,k} - p'_H \operatorname{div}\boldsymbol{v}_{T,k}\right) \end{aligned} \tag{3.179}$$

Denoting for any $F \in \mathcal{E}(T)$, $j = 1, \cdots, s$ and $k = 1, \cdots, r$

$$\begin{aligned} \lambda_{F,j} &= \int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{F,j} - p'_H \operatorname{div}\boldsymbol{v}_{F,j} \\ \lambda_{T,k} &= \int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v}_{T,k} - p'_H \operatorname{div}\boldsymbol{v}_{T,k} \end{aligned}$$

and taking advantage of $\beta_{E,i}$ (3.176) and $\beta_{T,l}$ (3.177), equation (3.179) can be written as

$$\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - p'_H \operatorname{div}\boldsymbol{v}\right) = \sum_{E\in\mathcal{E}(T)}\sum_{i=1}^{s}\lambda_{E,i}\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{E,i} + \sum_{k=1}^{r}\lambda_{T,k}\int_{T\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\varphi}_{T,k} \tag{3.180}$$

In (3.180), taking $\boldsymbol{v} = \mathbf{0}$ on $E \in \mathcal{E}(T)$ and integrating by parts the left hand side, we obtain

$$-\mu\Delta\boldsymbol{u}_H + \nabla p'_H = \sum_{k=1}^{r}\lambda_{T,k}\boldsymbol{\varphi}_{T,k} \text{ in } T\cap\Omega^\varepsilon \tag{3.181}$$

Then by writing the variational formulation of (3.181) for $\boldsymbol{v} \in V(T)$ and comparing with (3.180), it is easy to deduce that

$$\mu\nabla\boldsymbol{u}_H\boldsymbol{n} - p'_H\boldsymbol{n} \in \operatorname{span}\left\{\boldsymbol{\omega}_{E,1}, \cdots, \boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon\ \forall E\in\mathcal{E}(T) \tag{3.182}$$

Finally, combining equations (3.174), (3.181) and (3.182), we obtain the following system

$$\begin{aligned}
-\mu\Delta \boldsymbol{u}_H + \nabla p'_H &= \sum_{k=1}^{r} \lambda_{T,k}\boldsymbol{\varphi}_{T,k} \text{ in } T\cap\Omega^\varepsilon\\
\operatorname{div} \boldsymbol{u}_H &\in \operatorname{span}\left\{\varpi_{T,1},\cdots,\varpi_{T,t}\right\} \text{ in } T\cap\Omega^\varepsilon\\
\boldsymbol{u}_H &= 0 \text{ on } \partial B^\varepsilon\cap T\\
\mu\nabla \boldsymbol{u}_H\boldsymbol{n} - p'_H\boldsymbol{n} &\in \operatorname{span}\left\{\boldsymbol{\omega}_{E,1},\cdots,\boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E\in\mathcal{E}(T)
\end{aligned}$$

On any element $T \in \mathcal{T}_H$, for any $\boldsymbol{u}_H \in \widehat{V}_H^{St}$ fixed in the formulation above, we now prove that p'_H and the vector $(\lambda_{T,k}) = (\lambda_{T,1},\cdots,\lambda_{T,r})$ are both uniquely determined by $\boldsymbol{u}_H$. Assuming that there exist $\left(\zeta_1^T,\left(\lambda_{T,k}^1\right)\right) \in \widehat{M}_H^0(T)\times\mathbb{R}^r$ and $\left(\zeta_2^T,\left(\lambda_{T,k}^2\right)\right) \in \widehat{M}_H^0(T)\times\mathbb{R}^r$ that both verify the formulation above. We recall that the space $\widehat{M}_H^0(T)$ is defined by (3.164). In other words,

$$\begin{aligned}
\nabla\zeta_1^T &= \sum_{k=1}^{r} \lambda_{T,k}^1\boldsymbol{\varphi}_{T,k} + \mu\Delta\boldsymbol{u}_H\\
\nabla\zeta_2^T &= \sum_{k=1}^{r} \lambda_{T,k}^2\boldsymbol{\varphi}_{T,k} + \mu\Delta\boldsymbol{u}_H
\end{aligned}$$

Subtracting these equations, we obtain

$$\nabla\left(\zeta_1^T-\zeta_2^T\right) = \sum_{k=1}^{r}\left(\lambda_{T,k}^1-\lambda_{T,k}^2\right)\boldsymbol{\varphi}_{T,k}$$

The fact $\boldsymbol{\varphi}_{T,k} \in \left(\mathbb{P}_{n-1}(T)\right)^d$ implies that $\left(\zeta_1^T-\zeta_2^T\right) \in \mathbb{P}_n(T)$. Besides, $\left(\zeta_1^T-\zeta_2^T\right) \in \widehat{M}_H^0(T)$ which is orthogonal to $\mathbb{P}_n(T)$. Thus we deduce that $\zeta_1^T = \zeta_2^T$ and then $\left(\lambda_{T,k}^1\right) = \left(\lambda_{T,k}^2\right)$. Therefore we conclude that both p'_H and $(\lambda_{T,k})$ are uniquely determined by $\boldsymbol{u}_H$. Consequently, we deduce that there exists a linear operator such that $p'_H = \pi_H(\boldsymbol{u}_H)$ with $\boldsymbol{u}_H \in \widehat{V}_H^{St}$. We recall that the pressure p_H is defined by $p_H = \bar{p}_H + p'_H$. Thus we proved that $(\boldsymbol{u}_H, p_H)$ defined in (3.165) belongs to the space defined by (3.168).

Reciprocally, we now prove that $(\boldsymbol{u}_H, p_H) \in \widehat{\underline{X}}_H^{St}$ defined by (3.168) satisfies also relation (3.165). Let $(\boldsymbol{u}_H, p_H) \in \widehat{\underline{X}}_H^{St}$ defined by (3.168). For any $\boldsymbol{u}_H \in \widehat{V}_H^{St}$, on any $T \in \mathcal{T}_H$, definition of $\widehat{V}_H^{St}$ (3.162) shows that there exist a unique $\zeta^T = \pi_H(\boldsymbol{u}_H) \in \widehat{M}_H^0(T)$ and a unique $(\lambda_{T,1},\cdots,\lambda_{T,r}) \in \mathbb{R}^r$ such that

$$\begin{aligned}
-\mu\Delta \boldsymbol{u}_H + \nabla \zeta^T &= \sum_{k=1}^{r} \lambda_{T,k}\boldsymbol{\varphi}_{T,k} \text{ in } T\cap\Omega^\varepsilon && (3.183)\\
\operatorname{div} \boldsymbol{u}_H &\in \operatorname{span}\left\{\varpi_{T,1},\cdots,\varpi_{T,t}\right\} \text{ in } T\cap\Omega^\varepsilon && (3.184)\\
\boldsymbol{u}_H &= \boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T && (3.185)\\
\mu\nabla \boldsymbol{u}_H\boldsymbol{n} - \zeta^T\boldsymbol{n} &\in \operatorname{span}\left\{\boldsymbol{\omega}_{E,1},\cdots,\boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon,\ \forall E\in\mathcal{E}(T) && (3.186)
\end{aligned}$$

where $\pi_H : \widehat{V}_H^{St} \to \widehat{M}_H^0$ is the linear operator in Definition 3.6.4.

Equation (3.186) shows that for any $E \in \mathcal{E}(T)$, there exist $\lambda_{E,1}, \cdots, \lambda_{E,s} \in \mathbb{R}$ such that

$$\mu \nabla \boldsymbol{u}_H \boldsymbol{n} - \zeta^T \boldsymbol{n} = \sum_{i=1}^{s} \lambda_{E,i} \boldsymbol{\omega}_{E,i} \text{ on } E \cap \Omega^\varepsilon,\ \forall E \in \mathcal{E}(T)$$

For any $\boldsymbol{v} \in \widehat{V}_H^0$, integrating by parts (3.183) and making use of the equation above, we obtain

$$\begin{aligned} \int_{T\cap\Omega^\varepsilon} \mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} &= \sum_{k=1}^{r} \lambda_{T,k} \int_{T\cap\Omega^\varepsilon} \boldsymbol{\varphi}_{T,k} \cdot \boldsymbol{v} \\ &+ \sum_{E\in\mathcal{E}(T)} \sum_{i=1}^{s} \lambda_{E,i} \int_{E\cap\Omega^\varepsilon} \boldsymbol{\omega}_{E,i} \cdot \boldsymbol{v} \end{aligned} \tag{3.187}$$

For any $\boldsymbol{v} \in \widehat{V}_H^0$, definition of $\widehat{V}_H^0$ (3.162) implies that

$$\int_{T\cap\Omega^\varepsilon} \boldsymbol{\varphi}_{T,k} \cdot \boldsymbol{v} = 0, \quad \int_{E\cap\Omega^\varepsilon} \boldsymbol{\omega}_{E,i} \cdot \boldsymbol{v} = 0$$

Combining equations above, it is easy to see that (3.187) reduces to

$$\int_{T\cap\Omega^\varepsilon} \mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \zeta^T \text{ div } \boldsymbol{v} = 0 \tag{3.188}$$

Besides, integration by parts yields

$$\int_{T\cap\Omega^\varepsilon} \bar{p}_H|_T \text{ div } \boldsymbol{v} = \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v} \cdot \boldsymbol{n} \bar{p}_H|_T - \int_{T\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \text{ grad } \bar{p}_H|_T,\ \forall \boldsymbol{v} \in \widehat{V}_H^0$$

The definition of $\widehat{M}_H$ implies that $\bar{p}_H|_T \in \mathbb{P}_n(T)$ and then grad $\bar{p}_H|_T \in (\mathbb{P}_{n-1}(T))^d$. It is also obvious that the restriction of $\boldsymbol{n}\bar{p}_H|_T$ to $E \in \mathcal{E}(T)$ belongs to $(\mathbb{P}_n(E))^d$. Making use of Assumption 3.1.1 and Assumption 3.6.2, definition of $\widehat{V}_H^0$ implies that

$$\begin{cases} \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v} \cdot \boldsymbol{n} \bar{p}_H|_T = 0 \\ \int_{T\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \text{ grad } \bar{p}_H|_T = 0 \end{cases} \implies \int_{T\cap\Omega^\varepsilon} \bar{p}_H|_T \text{ div } \boldsymbol{v} = 0 \tag{3.189}$$

Moreover, for any $\boldsymbol{u}_H \in \widehat{V}_H^{St}$, div $\boldsymbol{u}_H \in \mathbb{P}_n(T)$. For any $q \in \widehat{M}_H^0(T)$, the fact that $\mathbb{P}_n(T)$ and $\widehat{M}_H^0(T)$ are orthogonal implies that

$$\int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0 \tag{3.190}$$

As a result, summing (3.188), (3.189) and (3.190), we obtain that $\forall\, (\boldsymbol{v}, q) \in \widehat{X}_H^0$,

$$\int_{T\cap\Omega^\varepsilon} \mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \left(\zeta^T + \bar{p}_H|_T\right) \text{ div } \boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0,\ \forall T \in \mathcal{T}_H$$

Denoting $p_H|_T = \zeta^T + \bar{p}_H|_T$ on each element $T \in \mathcal{T}_H$, let p_H be the function which equals to $p_H|_T$ on each $T \in \mathcal{T}_H$, then $(\boldsymbol{u}_H, p_H) \in \widehat{X}_H^{St}$. Summing the equation above on all elements

$T \in \mathcal{T}_H$, we obtain that for $(\boldsymbol{v}, q) \in \widehat{X}_H^0$,

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} p_H \operatorname{div}\boldsymbol{v} - \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} q \operatorname{div}\boldsymbol{u}_H = 0$$

which is exactly definition (3.165).

Consequently, we have proved the identity between (3.165) and (3.168), i.e. property (3.169). □

3.6.2 The local problems defined by Stokes equations

We now construct a basis of the space $\widehat{V}_H^{St}$ which consists of functions associated to coarse elements (element-based basis functions) or faces (face-based basis functions) of the coarse mesh.

The strong form We first construct basis functions associated to faces of the coarse mesh. For any $E \in \mathcal{E}_H$, for $i = 1, \cdots, s$, find the function $\boldsymbol{\Phi}_{E,i} : \Omega^\varepsilon \to \mathbb{R}^d$, the pressure $\pi_{E,i} : \Omega^\varepsilon \to \mathbb{R}$ such that $\boldsymbol{\Phi}_{E,i}$ and $\pi_{E,i}$ vanish outside the coarse element $T_k \subset \omega_E$ for $k \in \{1, 2\}$ (only one coarse element if $E \in \partial\Omega$) and solve on T_k:

$$\begin{cases} -\mu\Delta\boldsymbol{\Phi}_{E,i} + \nabla\pi_{E,i} \in \operatorname{span}\{\boldsymbol{\varphi}_{T_k,1}, \cdots, \boldsymbol{\varphi}_{T_k,r}\} \text{ in } T_k \cap \Omega^\varepsilon \\ \operatorname{div}\boldsymbol{\Phi}_{E,i} \in \operatorname{span}\{\varpi_{T_k,1}, \cdots, \varpi_{T_k,t}\} \text{ in } T_k \cap \Omega^\varepsilon \\ \mu\nabla\boldsymbol{\Phi}_{E,i}\boldsymbol{n} - \pi_{E,i}\boldsymbol{n} \in \operatorname{span}\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\} \text{ on } F,\ \forall F \in \mathcal{E}(T_k) \\ \boldsymbol{\Phi}_{E,i} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T_k \\ \int_{F\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\omega}_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases} \quad \forall F \in \mathcal{E}(T_k),\ \forall j = 1, \cdots, s \\ \int_{T_k\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\varphi}_{T_k,l} = 0\ \forall l = 1, \cdots, r \\ \int_{T_k\cap\Omega^\varepsilon}\pi_{E,i}\cdot\varpi_{T_k,m} = 0\ \forall m = 1, \cdots, t \end{cases} \tag{3.191}$$

Now we construct basis functions associated to elements of the coarse mesh. For each $T \in \mathcal{T}_H$, for $k = 1, \cdots, r$, the support of the function $\boldsymbol{\Psi}_{T,k}$ is reduced to $T \cap \Omega^\varepsilon$. We find $\boldsymbol{\Psi}_{T,k} : \Omega^\varepsilon \to \mathbb{R}^d$ and $\pi_{T,k} : \Omega^\varepsilon \to \mathbb{R}$ by solving on T:

$$\begin{cases} -\mu\Delta\boldsymbol{\Psi}_{T,k} + \nabla\pi_{T,k} \in \operatorname{span}\{\boldsymbol{\varphi}_{T,1}, \cdots, \boldsymbol{\varphi}_{T,r}\} \text{ in } T \cap \Omega^\varepsilon \\ \operatorname{div}\boldsymbol{\Psi}_{T,k} \in \operatorname{span}\{\varpi_{T,1}, \cdots, \varpi_{T,t}\} \text{ in } T \cap \Omega^\varepsilon \\ \mu\nabla\boldsymbol{\Psi}_{T,k}\boldsymbol{n} - \pi_{T,k}\boldsymbol{n} \in \operatorname{span}\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\} \text{ on } F,\ \forall F \in \mathcal{E}(T) \\ \boldsymbol{\Psi}_{T,k} = 0 \text{ on } \partial B^\varepsilon \cap T \\ \int_{F\cap\Omega^\varepsilon}\boldsymbol{\Psi}_{T,k}\cdot\boldsymbol{\omega}_{F,j} = 0\ \forall F \in \mathcal{E}(T),\ \forall j = 1, \cdots, s \\ \int_{T\cap\Omega^\varepsilon}\boldsymbol{\Psi}_{T,k}\cdot\boldsymbol{\varphi}_{T,l} = \delta_{kl}\ \forall l = 1, \cdots, r \\ \int_{T\cap\Omega^\varepsilon}\pi_{T,k}\cdot\varpi_{T,m} = 0\ \forall m = 1, \cdots, t \end{cases} \tag{3.192}$$

The weak form The weak form of (3.191) is: for any $E \in \mathcal{E}_H$, for $i = 1, \cdots, s$, on the coarse element $T_k \subset \omega_E$ for $k \in \{1, 2\}$ (only one coarse element if $E \in \partial\Omega$), find $\boldsymbol{\Phi}_{E,i} \in \left(H^1\left(T_k \cap \Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{\Phi}_{E,i} = \boldsymbol{0}$ on $\partial B^\varepsilon \cap T_k$, $\pi_{E,i} \in L_0^2\left(T_k \cap \Omega^\varepsilon\right)$, $\lambda_{F,1}, \cdots, \lambda_{F,s} \in \mathbb{R}$

for $F \in \mathcal{E}\left(T_k\right)$ and $\lambda_{T_k,1}, \cdots, \lambda_{T_k,r} \in \mathbb{R}$ by solving:

$$\begin{cases} \int_{T_k\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{\Phi}_{E,i} : \nabla\boldsymbol{v} - \int_{T_k\cap\Omega^\varepsilon} \pi_{E,i}\,\text{div}\,\boldsymbol{v} + \sum\limits_{F\in\mathcal{E}(T_k)}\sum\limits_{j=1}^{s} \lambda_{F,j}\int_{F\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{F,j} = \\ \sum\limits_{l=1}^{r} \lambda_{T_k,l}\int_{T_k\cap\Omega^\varepsilon}\boldsymbol{\varphi}_{T_k,l}\cdot\boldsymbol{v},\ \forall\boldsymbol{v}\in\left(H^1\left(T_k\cap\Omega^\varepsilon\right)\right)^d \text{ such that } \boldsymbol{v}=\boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T_k \\ \int_{T_k\cap\Omega^\varepsilon} q\,\text{div}\,\boldsymbol{\Phi}_{E,i} = 0,\ \forall q\in\widehat{M}_H^0(T) \\ \sum\limits_{F\in\mathcal{E}(T_k)}\sum\limits_{j=1}^{s}\mu_{F,j}\int_{F\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\omega}_{F,j} = \mu_{E,i},\ \forall\mu_{F,j}\in\mathbb{R},\ \forall F\in\mathcal{E}(T_k),\ \forall j=1,\cdots,s \\ \int_{T_k\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\varphi}_{T_k,l} = 0,\ \forall l=1,\cdots,r \\ \int_{T_k\cap\Omega^\varepsilon}\pi_{E,i}\cdot\varpi_{T_k,m} = 0,\ \forall m=1,\cdots,t \end{cases}$$

The weak form of (3.192) is: for any $T \in \mathcal{T}_H$, for $k = 1, \cdots, r$, find $\boldsymbol{\Psi}_{T,k} \in \left(H^1\left(T\cap\Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{\Psi}_{T,k} = \boldsymbol{0}$ on $\partial B^\varepsilon \cap T$, $\pi_{T,k} \in L_0^2\left(T\cap\Omega^\varepsilon\right)$, $\lambda_{F,1}, \cdots, \lambda_{F,s} \in \mathbb{R}$ for all $F \in \mathcal{E}\left(T\right)$ and $\lambda_{T,1}, \cdots, \lambda_{T,r} \in \mathbb{R}$ by solving

$$\begin{cases} \int_{T\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{\Psi}_{T,k} : \nabla\boldsymbol{v} - \int_{T\cap\Omega^\varepsilon} \pi_{T,k}\,\text{div}\,\boldsymbol{v} + \sum\limits_{F\in\mathcal{E}(T)}\sum\limits_{j=1}^{s} \lambda_{F,j}\int_{F\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{F,j} = \\ \sum\limits_{l=1}^{r} \lambda_{T,l}\int_{T\cap\Omega^\varepsilon}\boldsymbol{\varphi}_{T,l}\cdot\boldsymbol{v},\ \forall\boldsymbol{v}\in\left(H^1\left(T\cap\Omega^\varepsilon\right)\right)^d \text{ such that } \boldsymbol{v}=\boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T \\ \int_{T\cap\Omega^\varepsilon} q\,\text{div}\,\boldsymbol{\Psi}_{T,k} = 0, \forall q\in\widehat{M}_H^0(T) \\ \sum\limits_{F\in\mathcal{E}(T)}\sum\limits_{j=1}^{s}\mu_{F,j}\int_{F\cap\Omega^\varepsilon}\boldsymbol{\Psi}_{T,k}\cdot\boldsymbol{\omega}_{F,j} = 0,\ \forall\mu_{F,j}\in\mathbb{R},\ \forall F\in\mathcal{E}(T),\ \forall j=1,\cdots,s \\ \int_{T\cap\Omega^\varepsilon}\boldsymbol{\Psi}_{T,k}\cdot\boldsymbol{\varphi}_{T,l} = \delta_{kl},\ \forall l=1,\cdots,r \\ \int_{T\cap\Omega^\varepsilon}\pi_{T,k}\cdot\varpi_{T,m} = 0,\ \forall m=1,\cdots,t \end{cases}$$

3.6.3 The basis functions of the space $\widehat{V}_H^{St}$

Theorem 3.6.2. *The functions* $\boldsymbol{\Phi}_{E,i}$ *for* $E \in \mathcal{E}_H$ *and* $i = 1, \cdots, s$ *defined by* (3.191) *and* $\boldsymbol{\Psi}_{T,k}$ *for* $T \in \mathcal{T}_H$ *and* $k = 1, \cdots, r$ *defined by* (3.192) *form a basis of* $\widehat{V}_H^{St}$ *defined by* (3.167)*. In other words,*

$$\widehat{V}_H^{St} = \text{span}\left\{\boldsymbol{\Phi}_{E,i},\ \boldsymbol{\Psi}_{T,k},\ E\in\mathcal{E}_H,\ T\in\mathcal{T}_H,\ i=1,\cdots,s,\ k=1,\cdots,r\right\} \tag{3.193}$$

This theorem is new and now we provide its detailed proof in what follows.

Proof. First of all, it is obvious that functions $\boldsymbol{\Phi}_{E,i}$, for $E \in \mathcal{E}_H$ and $i = 1, \cdots, s$ defined by (3.191) and $\boldsymbol{\Psi}_{T,k}$ for $T \in \mathcal{T}_H$ and $k = 1, \cdots, r$ defined by (3.192) belong both to $\widehat{V}_H^{St}$. Besides, it is easy to verify that $\left\{\boldsymbol{\Phi}_{E,i},\ E\in\mathcal{E}_H,\ i=1,\cdots,s\right\} \cup \left\{\boldsymbol{\Psi}_{T,k},\ T\in\mathcal{T}_H,\ k=1,\cdots,r\right\}$ forms a linearly independent family. Consequently, we have

$$\text{span}\left\{\boldsymbol{\Phi}_{E,i},\ \boldsymbol{\Psi}_{T,k},\ E\in\mathcal{E}_H,\ T\in\mathcal{T}_H,\ i=1,\cdots,s,\ k=1,\cdots,r\right\} \subset \widehat{V}_H^{St}$$

Reciprocally, let $\boldsymbol{u} \in \widehat{V}_H^{St}$, definition of $\widehat{V}_H^{St}$ implies that on each $T \in \mathcal{T}_H$, there exist a unique $\zeta^T \in M_H^0(T)$ and a unique vector $\left(\lambda_{T,1}, \cdots, \lambda_{T,r}\right) \in \mathbb{R}^r$ that satisfy (3.183)–(3.186).

We introduce $\boldsymbol{v}$ and σ below:

$$\boldsymbol{v} = \boldsymbol{u} - \sum_{T\in\mathcal{T}_H}\sum_{k=1}^{r}\left(\int_{T\cap\Omega^\varepsilon}\boldsymbol{u}\cdot\boldsymbol{\varphi}_{T,k}\right)\boldsymbol{\Psi}_{T,k} - \sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s}\left(\int_{E\cap\Omega^\varepsilon}\boldsymbol{u}\cdot\boldsymbol{\omega}_{E,i}\right)\boldsymbol{\Phi}_{E,i}$$

$$\sigma = \zeta^T - \sum_{T\in\mathcal{T}_H}\sum_{k=1}^{r}\left(\int_{T\cap\Omega^\varepsilon}\boldsymbol{u}\cdot\boldsymbol{\varphi}_{T,k}\right)\pi_{T,k} - \sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s}\left(\int_{E\cap\Omega^\varepsilon}\boldsymbol{u}\cdot\boldsymbol{\omega}_{E,i}\right)\pi_{E,i}$$

For any $T\in\mathcal{T}_H$, it is easy to check that $\boldsymbol{v}$ and σ verify:

$$-\mu\Delta\boldsymbol{v} + \nabla\sigma \in \text{span}\left\{\boldsymbol{\varphi}_{T,1},\cdots,\boldsymbol{\varphi}_{T,r}\right\} \text{ in } T\cap\Omega^\varepsilon \tag{3.194}$$

$$\text{div } \boldsymbol{v} \in \text{span}\left\{\varpi_{T,1},\cdots,\varpi_{T,t}\right\} \text{ in } T\cap\Omega^\varepsilon \tag{3.195}$$

$$\mu\nabla\boldsymbol{v}\boldsymbol{n} - \sigma\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{F,1},\cdots,\boldsymbol{\omega}_{F,s}\right\} \text{ on } F\cap\Omega^\varepsilon,\ \forall F\in\mathcal{E}(T) \tag{3.196}$$

$$\boldsymbol{v} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T$$

$$\int_{E\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\omega}_{E,i} = 0\ \forall E\in\mathcal{E}(T),\ \forall i = 1,\cdots,s \tag{3.197}$$

$$\int_{T\cap\Omega^\varepsilon}\boldsymbol{v}\cdot\boldsymbol{\varphi}_{T,l} = 0\ \forall l = 1,\cdots,r \tag{3.198}$$

$$\int_{T\cap\Omega^\varepsilon}\sigma\cdot\varpi_{T,j} = 0\ \forall j = 1,\cdots,t \tag{3.199}$$

Equation (3.194) implies that there exists $(\lambda_{T,1},\cdots,\lambda_{T,r})\in\mathbb{R}^r$ such that

$$-\mu\Delta\boldsymbol{v} + \nabla\sigma = \sum_{l=1}^{r}\lambda_{T,l}\boldsymbol{\varphi}_{T,l} \tag{3.200}$$

Choosing $\boldsymbol{v}$ as the test function, the variational formulation of (3.200) is

$$\int_{T\cap\Omega^\varepsilon}\mu|\nabla\boldsymbol{v}|^2 - \int_{T\cap\Omega^\varepsilon}\sigma \text{ div } \boldsymbol{v} = \sum_{l=1}^{r}\lambda_{T,l}\int_{T\cap\Omega^\varepsilon}\boldsymbol{\varphi}_{T,l}\cdot\boldsymbol{v} + \int_{\partial(T\cap\Omega^\varepsilon)}(\mu\nabla\boldsymbol{v}\boldsymbol{n} - \sigma\boldsymbol{n})\cdot\boldsymbol{v} \tag{3.201}$$

It is straightforward to deduce from (3.195) and (3.199) that

$$\int_{T\cap\Omega^\varepsilon}\sigma \text{ div } \boldsymbol{v} = 0$$

Then equation (3.198) reveals that

$$\sum_{l=1}^{r}\lambda_{T,l}\int_{T\cap\Omega^\varepsilon}\boldsymbol{\varphi}_{T,l}\cdot\boldsymbol{v} = 0$$

Combining equations (3.196) and (3.197), we can deduce that

$$\int_{\partial(T\cap\Omega^\varepsilon)}(\mu\nabla\boldsymbol{v}\boldsymbol{n} - \sigma\boldsymbol{n})\cdot\boldsymbol{v} = 0$$

Finally, equation (3.201) reduces to

$$\int_{T\cap\Omega^\varepsilon}|\nabla\boldsymbol{v}|^2 = 0$$

which implies that $\boldsymbol{v}$ is constant on $T \cap \Omega^\varepsilon$. On combining (3.198), we deduce that $\boldsymbol{v} = \mathbf{0}$ and thus $\sigma = 0$. Thus we have proved that

$$\widehat{V}_H^{St} \subset \operatorname{span}\left\{\boldsymbol{\Phi}_{E,i}, \boldsymbol{\Psi}_{T,k},\ E \in \mathcal{E}_H,\ T \in \mathcal{T}_H,\ i = 1, \cdots, s,\ k = 1, \cdots, r\right\}$$

Consequently, combing the results above, we have proved that

$$\widehat{V}_H^{St} = \operatorname{span}\left\{\boldsymbol{\Phi}_{E,i}, \boldsymbol{\Psi}_{T,k},\ E \in \mathcal{E}_H,\ T \in \mathcal{T}_H,\ i = 1, \cdots, s,\ k = 1, \cdots, r\right\}$$

☐

We conclude that any function $\widehat{V}_H^{St}$ can be represented as

$$\forall \boldsymbol{u}_H \in \widehat{V}_H^{St},\ \boldsymbol{u}_H = \sum_{E \in \mathcal{E}_H} \sum_{i=1}^{s} u_{E,i} \boldsymbol{\Phi}_{E,i} + \sum_{T \in \mathcal{T}_H} \sum_{k=1}^{r} u_{T,k} \boldsymbol{\Psi}_{T,k}$$

An explicit formulation of the linear operator π_H (see Definition 3.6.4) is

$$\forall \boldsymbol{u}_H \in \widehat{V}_H^{St},\ \pi_H\left(\boldsymbol{u}_H\right) = \sum_{E \in \mathcal{E}_H} \sum_{i=1}^{s} u_{E,i} \pi_{E,i} + \sum_{T \in \mathcal{T}_H} \sum_{k=1}^{r} u_{T,k} \pi_{T,k} \tag{3.202}$$

3.6.4 The choices of weighting functions and finite elements

For $n = 0$, weighting functions are chosen as (3.161) and thus the following condition

$$\int_{T \cap \Omega^\varepsilon} \boldsymbol{u} \cdot \boldsymbol{\varphi}_{T,k} = 0,\ \forall T \in \mathcal{T}_H,\ \forall k = 1, \cdots, r$$

is void in $\widehat{V}_H^0$ defined by (3.162).

For $n = 0$, Assumption 3.6.1 reveals that the choice of weighting functions (3.161) is the same as in the case of $s = 2$ defined by (3.7) and the case of $s = 3$ defined by (3.9). Thus the multiscale method defined by Stokes equations (see section 3.2) with weighting functions in the case of $s = 2$ defined by (3.7) and the case of $s = 3$ defined by (3.9) is a special case of the high-order multiscale method.

For $n = 1$, in two-dimensional simulations performed in this thesis, we choose for any $T \in \mathcal{T}_H$, for any $E \in \mathcal{E}_H$,

$$\begin{cases} s = 4: \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2,\ \boldsymbol{\omega}_{E,3} = \boldsymbol{n}_E \psi_E,\ \boldsymbol{\omega}_{E,4} = \boldsymbol{\tau}_E \phi_E. \\ t = 3: \ \varpi_{T,1} = 1,\ \varpi_{T,2} = x,\ \varpi_{T,3} = y. \\ r = 2: \ \boldsymbol{\varphi}_{T,1} = \boldsymbol{e}_1,\ \boldsymbol{\varphi}_{T,2} = \boldsymbol{e}_2. \end{cases} \tag{3.203}$$

where $\{\boldsymbol{e}_1,\ \boldsymbol{e}_2\}$ is the canonical basis of $\mathbb{R}^2$ and $\boldsymbol{n}_E$ and $\boldsymbol{\tau}_E$ are respectively unit vectors normal and tangent to the face E. Besides, $\psi_E \in \mathbb{P}_1(E)$ and $\phi_E \in \mathbb{P}_1(E)$ which verify that $\int_{E \cap \Omega^\varepsilon} \psi_E = 0$ and $\int_{E \cap \Omega^\varepsilon} \phi_E = 0$. We have not performed three-dimensional simulations using high-order multiscale methods in this thesis. But the choice of weighting functions in three dimensions is straightforward.

In this case, by comparing $\widehat{V}_H^0$ and $\widehat{M}_H^0$ defined by (3.162)–(3.163) with $V_{H,bubble}^0$ and M_H^0 defined by (3.128)–(3.129), we find that the definition of the multiscale method enriched by bubble functions is not completely correct. The space $V_{H,bubble}^0$ defined by (3.128) is almost correctly defined whereas M_H^0 defined by (3.129) should be enriched by weighting functions $\boldsymbol{\varphi}_{T,k}$ for $T \in \mathcal{T}_H$ and $k = 1, \cdots, r$. This explains why this multiscale method does not improve the accuracy of numerical results. The high-order multiscale method with $n = 1$ gives the correct definition of bubble functions for Stokes equations and indicates that weighting functions should be chosen as (3.203). This definition of bubble functions is quite different from that for diffusion or advection-diffusion problems in [54, 102, 105, 113], due to the existence of pressure.

In conclusion, the high-order Crouzeix-Raviart multiscale finite element method provides a more general framework which provides appropriate definitions of $\widehat{V}_H^0$ and $\widehat{M}_H^0$ for different $n \geqslant 0$. For any $n \geqslant 1$ given, this method indicates that weighting functions should be chosen as Assumption 3.6.2. When n increases, the approximation space $\widehat{X}_H^{St}$ is enriched and the multiscale method becomes more accurate. However, at the same time, there are more weighting functions to solve and computing costs increase. By choosing an appropriate n, the high-order multiscale method allows to find a compromise between the expected accuracy and affordable computing costs.

Besides, numerical experiments performed in this thesis reveals that the finite element used to discretize local problems (3.191)–(3.192) plays an important role in the accuracy of the high-order Crouzeix-Raviart multiscale method. In practice, we first tried to use the Crouzeix-Raviart finite element to discretize local problems (3.191)–(3.192) with weighting functions chosen as (3.203). We recall that in the Crouzeix-Raviart finite element, the pressure is discretized in the $\mathbb{P}_0$ space and the velocity in the $\mathbb{P}_1$ nonconforming space. Then we tried also to discretize the pressure of local problems in the $\mathbb{P}_1$ space without changing the discretization of the velocity. Numerical results (see Chapter 6) reveals that the reconstructed fine-scale pressure is more accurate when we discretize the pressure of local problems in the $\mathbb{P}_1$ space.

3.6.5 The coarse-scale problem

The coarse-scale formulation of Stokes problem (1.2) reads: find $(\boldsymbol{u}_H, p_H) \in \widehat{X}_H^{St}$ such that

$$c_H^{St}\left((\boldsymbol{u}_H, p_H), (\boldsymbol{v}, q)\right) = (\boldsymbol{f}, \boldsymbol{v}), \ \forall (\boldsymbol{v}, q) \in \widehat{X}_H^{St}$$

Theorem 3.6.1 implies that p_H can be decomposed as $p_H = \pi_H(\boldsymbol{u}_H) + \bar{p}_H$ with $\pi_H(\boldsymbol{u}_H) \in \widehat{M}_H^0$ and $\bar{p}_H \in \widehat{M}_H$. It is easy to verify that $(\pi_H(\boldsymbol{u}_H), \operatorname{div} \boldsymbol{v}) = 0$ for all $\boldsymbol{u}_H, \boldsymbol{v} \in \widehat{V}_H^{St}$. Therefore, the problem above can be reformulated as: find $\boldsymbol{u}_H \in \widehat{V}_H^{St}$ and $\bar{p}_H \in \widehat{M}_H$ such that

$$a_H^{St}(\boldsymbol{u}_H, \boldsymbol{v}) + b_H(\boldsymbol{v}, \bar{p}_H) = F_H(\boldsymbol{v}), \ \forall \boldsymbol{v} \in \widehat{V}_H^{St} \tag{3.204}$$

$$b_H(\boldsymbol{u}_H, q) = 0, \ \forall q \in \widehat{M}_H \tag{3.205}$$

where

$$\begin{aligned} a_H^{St}(\boldsymbol{u}_H, \boldsymbol{v}) &= \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \mu \nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} \\ b_H(\boldsymbol{v}, \bar{p}_H) &= -\sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \bar{p}_H \operatorname{div} \boldsymbol{v} \\ F_H(\boldsymbol{v}) &= \sum_{T \in \mathcal{T}_H} \int_{T \cap \Omega^\varepsilon} \boldsymbol{f} \cdot \boldsymbol{v} \end{aligned}$$

Theorem 3.6.3. *The space $\widehat{V}_H^{St}$ and the space $\widehat{M}_H$ have the following relation*

$$div\, \widehat{V}_H^{St} = \widehat{M}_H$$

This theorem is new and now we provide its detailed proof in what follows.

Proof. We prove first that div $\widehat{V}_H^{St} \subset \widehat{M}_H$. For any $\boldsymbol{v} \in \widehat{V}_H^{St}$, for any $T \in \mathcal{T}_H$, we have proved that div $\boldsymbol{v} \in \mathbb{P}_n(T)$. As $\widehat{V}_H^{St} \subset \widehat{V}_H^{ext}$, definition of $\widehat{V}_H^{ext}$ implies that $\int_{E\cap\Omega^\varepsilon} [[\boldsymbol{v}]] \cdot \boldsymbol{n} = 0$. Thus we deduce that

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{v} = \sum_{T\in\mathcal{T}_H} \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v}\cdot\boldsymbol{n} = \sum_{E\in\mathcal{E}_H} \int_{E\cap\Omega^\varepsilon} [[\boldsymbol{v}\cdot\boldsymbol{n}]] = 0$$

Hence we have proved that div $\widehat{V}_H^{St} \subset \widehat{M}_H$.

Reciprocally, we now prove that $\widehat{M}_H \subset$ div $\widehat{V}_H^{St}$. For any $q \in \widehat{M}_H$, definition of $\widehat{M}_H$ shows that $q \in L_0^2(\Omega)$. Thus there exists $\boldsymbol{v} \in \left(H_0^1(\Omega^\varepsilon)\right)^d$ such that div $\boldsymbol{v} = q$. It is easy to check that $\boldsymbol{v}$ can be decomposed as

$$\boldsymbol{v} = \boldsymbol{v}_H + \boldsymbol{v}_H^0 \text{ with } \boldsymbol{v}_H \in \widehat{V}_H^{St},\ v_H^0 \in \widehat{V}_H^0 \tag{3.206}$$

For any element $T \in \mathcal{T}_H$, let $\varpi_T \in \mathbb{P}_n(T)$. Integrating by parts gives:

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \varpi_T \text{ div } \boldsymbol{v}_H = \sum_{T\in\mathcal{T}_H} \int_{\partial(T\cap\Omega^\varepsilon)} \varpi_T \boldsymbol{v}_H \cdot \boldsymbol{n} - \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \boldsymbol{v}_H \cdot \nabla\varpi_T$$

Substituting $\boldsymbol{v}_H$ defined in (3.206) into this equation, we obtain

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \varpi_T \text{ div } \boldsymbol{v}_H = \sum_{T\in\mathcal{T}_H} \int_{\partial(T\cap\Omega^\varepsilon)} \varpi_T \left(\boldsymbol{v} - \boldsymbol{v}_H^0\right) \cdot \boldsymbol{n} - \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\boldsymbol{v} - \boldsymbol{v}_H^0\right) \cdot \nabla\varpi_T \tag{3.207}$$

The fact that $\varpi_T \in \mathbb{P}_n(T)$ implies that $\boldsymbol{n}\varpi_T \in (\mathbb{P}_n(T))^d$ and $\nabla\varpi_T \in \mathbb{P}_{n-1}(T)$. Then definition of $\widehat{V}_H^0$ reveals that

$$\int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{v}_H^0 \cdot \boldsymbol{n}\varpi_T = 0, \quad \int_{T\cap\Omega^\varepsilon} \boldsymbol{v}_H^0 \cdot \nabla\varpi_T = 0$$

Finally (3.207) reduces to

$$\begin{aligned}
\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \varpi_T \text{ div } \boldsymbol{v}_H &= \sum_{T\in\mathcal{T}_H} \int_{\partial(T\cap\Omega^\varepsilon)} \varpi_T \boldsymbol{v} \cdot \boldsymbol{n} - \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \boldsymbol{v} \cdot \nabla\varpi_T \\
&= \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \varpi_T \text{ div } \boldsymbol{v} \\
&= \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} q\varpi_T
\end{aligned}$$

It is easy to deduce that $q =$ div $\boldsymbol{v}_H$ and thus $\widehat{M}_H \subset$ div $\widehat{V}_H^{St}$. Consequently, combining the results above, we conclude that div $\widehat{V}_H^{St} = \widehat{M}_H$. ☐

Making use of Theorem 3.6.3, it is trivial to deduce from (3.205) that div $\boldsymbol{u}_H = 0$ in $T \cap \Omega^\varepsilon$ for $T \in \mathcal{T}_H$. We can thus eliminate pressure from (3.204)–(3.205) by introducing the subspace:

$$\widehat{Z}_H^{St} = \left\{ \boldsymbol{v} \in \widehat{V}_H^{St} \text{ such that div } \boldsymbol{v} = 0,\ \forall T \in \mathcal{T}_H \right\}$$

Therefore (3.204)–(3.205) is equivalent to: find $\boldsymbol{u}_H \in \widehat{Z}_H^{St}$ such that

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v}=\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\boldsymbol{f}\cdot\boldsymbol{v},\ \forall\boldsymbol{v}\in\widehat{Z}_H^{St}$$

The existence and uniqueness of a velocity $\boldsymbol{u}_H$ to this problem is guaranteed by Theorem 2.1.2. Then the existence and uniqueness of pressure $\bar{p}_H$ follows from the fact that div $\widehat{V}_H^{St}=\widehat{M}_H$. As a result, we have proved that (3.204)–(3.205) has one and only one solution $(\boldsymbol{u}_H,\bar{p}_H)\in\widehat{V}_H^{St}\times\widehat{M}_H$.

Moreover, like other multiscale methods previously presented in this chapter, this high-order multiscale method can also be applied to solve Oseen problems and Navier-Stokes problems on the coarse mesh. The reader can refer to subsection 3.2.4 and subsection 3.3.5 for more details.

Remark. *In practice, we notice that it is not only inefficient but also difficult to apply the prediction-correction algorithm to solve local problems* (3.191)–(3.192) *and coarse problems* (3.204)–(3.205). *Thus an important work was devoted in this thesis to the implementation of a direct solver. More explanations about the choices of finite elements and solvers can be found in the next chapter.*

3.6.6 The reconstruction of fine-scale features

After solving the coarse-scale problems, we obtain the coarse-scale solutions $\boldsymbol{u}_H$ and $\bar{p}_H$. Then we reconstruct the fine-scale features of the solution for visualization. On each coarse element $T\in\mathcal{T}_H$, making use of $\widehat{V}_H^{St}$ defined by (3.193) and the formula of π_H defined by (3.202), the fine-scale velocity and pressure are approximated by

$$\begin{aligned}\boldsymbol{u}_H|_T&=\sum_{E\in\mathcal{E}(T)}\sum_{i=1}^{s}u_{E,i}\boldsymbol{\Phi}_{E,i}+\sum_{k=1}^{r}u_{T,k}\boldsymbol{\Psi}_{T,k}\\ p_H|_T&=\sum_{E\in\mathcal{E}(T)}\sum_{i=1}^{s}u_{E,i}\pi_{E,i}+\sum_{k=1}^{r}u_{T,k}\pi_{T,k}+\bar{p}_H|_T\end{aligned}\tag{3.208}$$

3.7 The high-order Crouzeix-Raviart multiscale finite element method defined by Oseen equations

The high-order Crouzeix-Raviart multiscale finite element method presented in the last section can be easily extended for Oseen equations. In this section, the choices of weighting functions and Assumption 3.6.1 and Assumption 3.6.2 are all valid. Besides, let functional spaces $\widehat{X}_H^{ext}$, $\widehat{V}_H^{ext}$, M, $\widehat{V}_H^{0}$ and $\widehat{M}_H^{0}$ be the same as those defined in section 3.6. Since this high-order Crouzeix-Raviart multiscale finite element method is defined by Oseen equations, we denote $\widehat{X}_H$ by $\widehat{X}_H^{Os}$ in order to distinguish from $\widehat{X}_H^{St}$.

Definition 3.7.1. *The subspace $\widehat{X}_H^{Os}$ is defined as the "orthogonal" complement of $\widehat{X}_H^{0}$ with respect to the bilinear form $\widehat{c}_H^{Os}(.,.)$:*

$$(\boldsymbol{u}_H,p_H)\in\widehat{X}_H^{Os}\Longleftrightarrow\widehat{c}_H^{Os}\left((\boldsymbol{u}_H,p_H),(\boldsymbol{v},q)\right)=0,\ \forall(\boldsymbol{v},q)\in\widehat{X}_H^{0}\tag{3.209}$$

and we recall:

$$\widehat{c}_H^{Os}((\boldsymbol{u}_H,p_H),(\boldsymbol{v},q)) = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v}+\frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{u}_H\right)\boldsymbol{v}-\frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{v}\right)\boldsymbol{u}_H\right) + \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(-p_H\, div\,\boldsymbol{v}-q\, div\,\boldsymbol{u}_H\right)$$

Definition 3.7.2. *Let the functional spaces* $\widehat{M}_H\subset M$ *and* $\widehat{V}_H^{Os}\subset\widehat{V}_H^{ext}$ *be defined by*

$$\widehat{M}_H = \left\{q\in L_0^2\left(\Omega\right) \text{ such that } q\mid_T\in\mathbb{P}_n(T),\ \forall T\in\mathcal{T}_H\right\} \tag{3.210}$$

$$\widehat{V}_H^{Os} = \left\{\begin{array}{l} \boldsymbol{v}\in\left(L^2\left(\Omega^\varepsilon\right)\right)^d : \forall T\in\mathcal{T}_H,\ \exists\zeta^T\in\widehat{M}_H^0(T) \text{ such that} \\ -\mu\Delta\boldsymbol{v}+\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{v}+\nabla\zeta^T\in\operatorname{span}\left\{\boldsymbol{\varphi}_{T,1},\cdots,\boldsymbol{\varphi}_{T,r}\right\} \text{ in } T\cap\Omega^\varepsilon \\ div\,\boldsymbol{v}\in\operatorname{span}\left\{\varpi_{T,1},\cdots,\varpi_{T,t}\right\} \text{ in } T\cap\Omega^\varepsilon \\ \boldsymbol{v}=\boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T \\ \mu\nabla\boldsymbol{v}\boldsymbol{n}-\frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\boldsymbol{n}\right)\boldsymbol{v}-\zeta^T\boldsymbol{n}\in\operatorname{span}\left\{\boldsymbol{\omega}_{E,1},\cdots,\boldsymbol{\omega}_{E,s}\right\} \text{ on } E\cap\Omega^\varepsilon\ \forall E\in\mathcal{E}(T) \end{array}\right\} \tag{3.211}$$

where $\mathcal{E}\left(T\right)$ *is the set of faces composing* ∂T*.*

Definition 3.7.3. *For any* $\boldsymbol{v}\in\widehat{V}_H^{Os}$*, on any element* $T\in\mathcal{T}_H$*, definition* (3.211) *implies that there exists a function* ζ^T *which is uniquely determined by* $\boldsymbol{v}$*. We can thus define a linear operator* $\pi_H:\widehat{V}_H^{Os}\to\widehat{M}_H^0$ *such that for any* $\boldsymbol{v}\in\widehat{V}_H^{Os}$*,* $\pi_H\left(\boldsymbol{v}\right)=\zeta^T$ *on any element* $T\in\mathcal{T}_H$*.*

Theorem 3.7.1. *Using* $\widehat{M}_H$ (3.210) *and* $\widehat{V}_H^{Os}$ (3.211)*, we define* $\underline{\widehat{X}}_H^{Os}$ *by*

$$\underline{\widehat{X}}_H^{Os} = \operatorname{span}\left\{\left(\boldsymbol{u}_H,\pi_H\left(\boldsymbol{u}_H\right)+\bar{p}_H\right),\boldsymbol{u}_H\in\widehat{V}_H^{Os},\ \bar{p}_H\in\widehat{M}_H\right\}$$

then the space $\widehat{X}_H^{Os}$ *defined in* (3.209) *satisfies the following property*

$$\widehat{X}_H^{Os} = \underline{\widehat{X}}_H^{Os} \tag{3.212}$$

The proof of this theorem is similar to that of subsection 3.3.1 and subsection 3.6.1. Thus we will not repeat it here and we apply this theorem directly in this thesis.

3.7.1 The local problems defined by Oseen equations

We now construct a basis for the space $\widehat{V}_H^{Os}$ which consists of functions associated to coarse elements (element-based basis functions) or faces (face-based basis functions) of the coarse mesh.

The strong form We first construct the basis function associated to faces of the coarse mesh. For any $E\in\mathcal{E}_H$, for $i=1,\cdots,s$, find $\boldsymbol{\Phi}_{E,i}:\Omega^\varepsilon\to\mathbb{R}^d$ such that $\boldsymbol{\Phi}_{E,i}=\boldsymbol{0}$ on $\partial B^\varepsilon\cap T_k$ and $\pi_{E,i}:\Omega^\varepsilon\to\mathbb{R}$ such that $\boldsymbol{\Phi}_{E,i}$ and $\pi_{E,i}$ vanish outside the coarse element $T_k\subset\omega_E$, $k\in\{1,2\}$

(only one coarse element if $E \in \partial\Omega$) and solve on T_k:

$$\left\{\begin{array}{l}
-\mu\Delta\boldsymbol{\Phi}_{E,i} + \rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{\Phi}_{E,i} + \nabla\pi_{E,i} \in \text{span}\left\{\boldsymbol{\varphi}_{T_k,1},\cdots,\boldsymbol{\varphi}_{T_k,r}\right\} \ \text{ in } T_k\cap\Omega^\varepsilon \\
\text{div }\boldsymbol{\Phi}_{E,i} \in \text{span}\left\{\varpi_{T_k,1},\cdots,\varpi_{T_k,t}\right\} \ \text{ in } T_k\cap\Omega^\varepsilon \\
\mu\nabla\boldsymbol{\Phi}_{E,i}\boldsymbol{n} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\boldsymbol{n}\right)\boldsymbol{\Phi}_{E,i} - \pi_{E,i}\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{F,1},\cdots,\boldsymbol{\omega}_{F,s}\right\} \ \text{ on } F\cap\Omega^\varepsilon,\ \forall F\in\mathcal{E}(T_k) \\
\boldsymbol{\Phi}_{E,i} = \boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T_k \\
\int_{F\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\omega}_{F,j} = \left\{\begin{array}{ll}\delta_{ij}, & F=E\\ 0, & F\neq E\end{array}\right. \quad \forall F\in\mathcal{E}\left(T_k\right),\ \forall j=1,\cdots,s \\
\int_{T_k\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\varphi}_{T_k,l} = 0\ \forall l=1,\cdots,r \\
\int_{T_k\cap\Omega^\varepsilon}\pi_{E,i}\cdot\varpi_{T_k,m} = 0\ \forall m=1,\cdots,t
\end{array}\right. \tag{3.213}$$

Now we construct the basis function associated to elements of the coarse mesh. For each $T\in\mathcal{T}_H$, for $k=1,\cdots,r$, the support of the function $\boldsymbol{\Psi}_{T,k}$ is reduced to $T\cap\Omega^\varepsilon$. We find $\boldsymbol{\Psi}_{T,k}:\Omega^\varepsilon\to\mathbb{R}^d$ and $\pi_{T,k}:\Omega^\varepsilon\to\mathbb{R}$ by solving on T:

$$\left\{\begin{array}{l}
-\mu\Delta\boldsymbol{\Psi}_{T,k} + \rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{\Psi}_{T,k} + \nabla\pi_{T,k} \in \text{span}\left\{\boldsymbol{\varphi}_{T,1},\cdots,\boldsymbol{\varphi}_{T,r}\right\} \ \text{ in } T\cap\Omega^\varepsilon \\
\text{div }\boldsymbol{\Psi}_{T,k} \in \text{span}\left\{\varpi_{T,1},\cdots,\varpi_{T,t}\right\} \ \text{ in } T\cap\Omega^\varepsilon \\
\mu\nabla\boldsymbol{\Psi}_{T,k}\boldsymbol{n} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\boldsymbol{n}\right)\boldsymbol{\Psi}_{T,k} - \pi_{T,k}\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{F,1},\cdots,\boldsymbol{\omega}_{F,s}\right\} \text{ on } F,\ \forall F\in\mathcal{E}(T) \\
\boldsymbol{\Psi}_{T,k} = \boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T \\
\int_{F\cap\Omega^\varepsilon}\boldsymbol{\Psi}_{T,k}\cdot\boldsymbol{\omega}_{F,j} = 0\ \forall F\in\mathcal{E}\left(T\right),\ \forall j=1,\cdots,s \\
\int_{T\cap\Omega^\varepsilon}\boldsymbol{\Psi}_{T,k}\cdot\boldsymbol{\varphi}_{T,l} = \delta_{kl}\ \forall l=1,\cdots,r \\
\int_{T\cap\Omega^\varepsilon}\pi_{T,k}\cdot\varpi_{T,m} = 0\ \forall m=1,\cdots,t
\end{array}\right. \tag{3.214}$$

The weak form In the weak form, the system (3.213) reads: for any $E\in\mathcal{E}\left(T_k\right)$, for $i=1,\cdots,s$, on the coarse element $T_k\subset\omega_E$, $k\in\{1,2\}$ (only one coarse element if $E\in\partial\Omega$), find $\boldsymbol{\Phi}_{E,i}\in\left(H^1(T_k\cap\Omega^\varepsilon)\right)^d$ such that $\boldsymbol{\Phi}_{E,i}=\boldsymbol{0}$, $\pi_{E,i}\in L^2_0\left(T_k\cap\Omega^\varepsilon\right)$, $\lambda_{F,1},\cdots,\lambda_{F,s}\in\mathbb{R}$ for $F\in\mathcal{E}(T_k)$ and $\lambda_{T_k,1},\cdots,\lambda_{T_k,r}\in\mathbb{R}$ such that

$$\left\{\begin{array}{l}
\int_{T_k\cap\Omega^\varepsilon}\mu\nabla\boldsymbol{\Phi}_{E,i}:\nabla\boldsymbol{v} + \int_{T_k\cap\Omega^\varepsilon}\left(\frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{v}\cdot\boldsymbol{\Phi}_{E,i}\right) \\
-\int_{T_k\cap\Omega^\varepsilon}\pi_{E,i}\text{div }\boldsymbol{v} + \sum\limits_{F\in\mathcal{E}(T_k)}\sum\limits_{j=1}^{s}\lambda_{F,j}\int_F\boldsymbol{v}\cdot\boldsymbol{\omega}_{F,j} = \sum\limits_{l=1}^{r}\lambda_{T_k,l}\int_{T_k\cap\Omega^\varepsilon}\boldsymbol{\varphi}_{T_k,l}\cdot\boldsymbol{v}, \\
\int_{T_k\cap\Omega^\varepsilon}q\text{ div }\boldsymbol{\Phi}_{E,i} = 0,\ \forall q\in\widehat{M^0_H}(T_k) \\
\sum\limits_{F\in\mathcal{E}(T_k)}\sum\limits_{j=1}^{s}\mu_{F,j}\int_F\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\omega}_{F,j} = \mu_{E,i},\ \forall\mu_{F,j}\in\mathbb{R},\ \forall F\in\mathcal{E}(T_k),\ \forall j=1,\cdots,s \\
\int_{T_k\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\varphi}_{T_k,l} = 0,\ \forall l=1,\cdots,r \\
\int_{T_k\cap\Omega^\varepsilon}\pi_{E,i}\cdot\varpi_{T_k,m} = 0,\ \forall m=1,\cdots,t
\end{array}\right.$$

for all $\boldsymbol{v}\in\left(H^1\left(T_k\cap\Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{v}=\boldsymbol{0}$ on $\partial B^\varepsilon\cap T_k$.

The weak form of (3.214) is: on any coarse element $T\in\mathcal{T}_H$, for $k=1,\cdots,r$, find $\boldsymbol{\Psi}_{T,k}\in\left(H^1\left(T\cap\Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{\Psi}_{T,k}=\boldsymbol{0}$ on $\partial B^\varepsilon\cap T$, $\pi_{T,k}\in L^2_0\left(T\cap\Omega^\varepsilon\right)$, $\lambda_{F,1},\cdots,\lambda_{F,s}\in\mathbb{R}$ for all $F \in \mathcal{E}(T)$ and $\lambda_{T,1}, \cdots, \lambda_{T,r} \in \mathbb{R}$ such that

$$\begin{cases} \int_{T\cap\Omega^\varepsilon} \mu \nabla \boldsymbol{\Psi}_{T,k} : \nabla \boldsymbol{v} + \int_{T\cap\Omega^\varepsilon} \left(\frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla\right) \boldsymbol{\Psi}_{T,k} \cdot \boldsymbol{v} - \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \nabla\right) \boldsymbol{v} \cdot \boldsymbol{\Psi}_{T,k}\right) \\ - \int_{T\cap\Omega^\varepsilon} \pi_{T,k} \operatorname{div} \boldsymbol{v} + \sum\limits_{F\in\mathcal{E}(T)} \sum\limits_{j=1}^{s} \lambda_{F,j} \int_F \boldsymbol{v} \cdot \boldsymbol{\omega}_{F,j} = \sum\limits_{l=1}^{r} \lambda_{T,l} \int_{T\cap\Omega^\varepsilon} \boldsymbol{\varphi}_{T,l} \cdot \boldsymbol{v}, \\ \int_{T\cap\Omega^\varepsilon} q \operatorname{div} \boldsymbol{\Psi}_{T,k} = 0, \forall q \in \widehat{M_H^0}(T) \\ \sum\limits_{F\in\mathcal{E}(T)} \sum\limits_{j=1}^{s} \mu_{F,j} \int_F \boldsymbol{\Psi}_{T,k} \cdot \boldsymbol{\omega}_{F,j} = 0, \ \forall \mu_{F,j} \in \mathbb{R}, \ \forall F \in \mathcal{E}(T), \ \forall j = 1, \cdots, s \\ \int_{T\cap\Omega^\varepsilon} \boldsymbol{\Psi}_{T,k} \cdot \boldsymbol{\varphi}_{T,l} = \delta_{kl}, \ \forall l = 1, \cdots, r \\ \int_{T\cap\Omega^\varepsilon} \pi_{T,k} \cdot \varpi_{T,m} = 0, \ \forall m = 1, \cdots, t \end{cases}$$

for all $\boldsymbol{v} \in \left(H^1\left(T \cap \Omega^\varepsilon\right)\right)^d$ such that $\boldsymbol{v} = \boldsymbol{0}$ on $\partial B^\varepsilon \cap T$.

3.7.2 The basis functions of the space $\widehat{V}_H^{Os}$

Theorem 3.7.2. *The functions $\boldsymbol{\Phi}_{E,i}$ for $E \in \mathcal{E}_H$ and $i = 1, \cdots, s$ defined by (3.213) and $\boldsymbol{\Psi}_{T,k}$ for $T \in \mathcal{T}_H$ and $k = 1, \cdots, r$ defined by (3.214) form a basis of $\widehat{V}_H^{Os}$ defined by (3.211). In other words,*

$$\widehat{V}_H^{Os} = \operatorname{span}\left\{\boldsymbol{\Phi}_{E,i}, \ \boldsymbol{\Psi}_{T,k}, \ E \in \mathcal{E}_H, \ T \in \mathcal{T}_H, \ i = 1, \cdots, s, \ k = 1, \cdots, r\right\} \tag{3.215}$$

The proof of this theorem is completely similar to that of subsection 3.6.3. We conclude that any function $\widehat{V}_H^{Os}$ can be represented as

$$\forall \boldsymbol{u}_H \in \widehat{V}_H^{Os}, \ \boldsymbol{u}_H = \sum_{E\in\mathcal{E}_H} \sum_{i=1}^{s} u_{E,i} \boldsymbol{\Phi}_{E,i} + \sum_{T\in\mathcal{T}_H} \sum_{k=1}^{r} u_{T,k} \boldsymbol{\Psi}_{T,k}$$

An explicit formulation of the linear operator π_H (see Definition 3.6.4) is

$$\forall \boldsymbol{u}_H \in \widehat{V}_H^{Os}, \ \pi_H\left(\boldsymbol{u}_H\right) = \sum_{E\in\mathcal{E}_H} \sum_{i=1}^{s} u_{E,i} \pi_{E,i} + \sum_{T\in\mathcal{T}_H} \sum_{k=1}^{r} u_{T,k} \pi_{T,k} \tag{3.216}$$

3.7.3 The choices of weighting functions and finite elements

For $n = 0$, weighting functions are chosen as (3.161) and the following condition

$$\int_{T\cap\Omega^\varepsilon} \boldsymbol{u} \cdot \boldsymbol{\varphi}_{T,k} = 0, \ \forall T \in \mathcal{T}_H, \ k = 1, \cdots, r$$

is void in $\widehat{V}_H^0$ defined by (3.162).

For $n = 0$, Assumption 3.6.1 reveals that the choices of weighting functions (3.161) are the same as in the case of $s = 2$ defined by (3.7) and the case of $s = 3$ defined by (3.9). Thus the multiscale method defined by Oseen equations (see section 3.3) with weighting functions chosen in the case of $s = 2$ defined by (3.7) and the case of $s = 3$ defined by (3.9) is just a special case of the high-order multiscale method.

For $n=1$, in two-dimensional numerical simulations performed in this thesis, we choose for any $T\in\mathcal{T}_H$, for any $E\in\mathcal{E}_H$,

$$\begin{cases} s=4: \ \boldsymbol{\omega}_{E,1}=\boldsymbol{e}_1, \ \boldsymbol{\omega}_{E,2}=\boldsymbol{e}_2, \ \boldsymbol{\omega}_{E,3}=\boldsymbol{n}_E\psi_E, \ \boldsymbol{\omega}_{E,4}=\boldsymbol{\tau}_E\phi_E. \\ t=3: \ \varpi_{T,1}=1, \ \varpi_{T,2}=x, \ \varpi_{T,3}=y. \\ r=2: \ \boldsymbol{\varphi}_{T,1}=\boldsymbol{e}_1, \ \boldsymbol{\varphi}_{T,2}=\boldsymbol{e}_2. \end{cases}$$

where $\{\boldsymbol{e}_1, \ \boldsymbol{e}_2\}$ is the canonical basis of $\mathbb{R}^2$ and $\boldsymbol{n}_E$ and $\boldsymbol{\tau}_E$ are respectively unit vectors normal and tangent to the face E. Besides, $\psi_E\in\mathbb{P}_1(E)$ and $\phi_E\in\mathbb{P}_1(E)$ which verify that $\int_{E\cap\Omega^\varepsilon}\psi_E=0$ and $\int_{E\cap\Omega^\varepsilon}\phi_E=0$. Three-dimensional numerical simulations are not performed with high-order multiscale method in this thesis. But the choice of weighting functions in three dimensions is straightforward.

3.7.4 The coarse-scale problem

The discrete variational formulation of Oseen problem (1.3) is: find $(\boldsymbol{u}_H,p_H)\in\widehat{X}_H^{Os}$ such that

$$c_H^{Os}\left(\left(\boldsymbol{u}_H,p_H\right),\left(\boldsymbol{v},q\right)\right)=\left(\boldsymbol{f},\boldsymbol{v}\right), \ \forall\left(\boldsymbol{v},q\right)\in\widehat{X}_H^{Os}$$

where

$$\begin{aligned} c_H^{Os}((\boldsymbol{u}_H,p_H),(\boldsymbol{v},q)) = &\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v}+\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H\cdot\boldsymbol{v}\right) \\ &+\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(-p_H \operatorname{div}\boldsymbol{v}-q\operatorname{div}\boldsymbol{u}_H\right) \end{aligned}$$

Theorem 3.7.1 shows that p_H can be decomposed as $p_H=\pi_H\left(\boldsymbol{u}_H\right)+\bar{p}_H$ with $\bar{p}_H\in\widehat{M}_H$. Besides, it is easy to check that $\left(\pi_H\left(\boldsymbol{u}_H\right),\operatorname{div}\boldsymbol{v}\right)=0$ for all $\boldsymbol{u}_H,\boldsymbol{v}\in\widehat{V}_H^{Os}$. Making use of this property, the formula above is equivalent to: find $\boldsymbol{u}_H\in\widehat{V}_H^{Os}$ and $\bar{p}_H\in\widehat{M}_H$ such that

$$a_H^{Os}\left(\boldsymbol{u}_H,\boldsymbol{v}\right)+b_H\left(\boldsymbol{v},\bar{p}_H\right)=F_H\left(\boldsymbol{v}\right), \ \forall\boldsymbol{v}\in\widehat{V}_H^{Os} \tag{3.217}$$

$$b_H\left(\boldsymbol{u}_H,q\right)=0, \ \forall q\in\widehat{M}_H \tag{3.218}$$

and we recall

$$a_H^{Os}\left(\boldsymbol{u}_H,\boldsymbol{v}\right)=\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H:\nabla\boldsymbol{v}+\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H\cdot\boldsymbol{v}\right)$$

Theorem 3.7.3. *The space $\widehat{V}_H^{Os}$ and the space $\widehat{M}_H$ satisfy the following relation*

$$\mathit{div}\ \widehat{V}_H^{Os}=\widehat{M}_H$$

The proof of this theorem is similar to that of subsection 3.6.5 and we will not repeat it here. Taking advantage of this theorem, we can deduce from (3.218) that $\operatorname{div}\boldsymbol{u}_H=0$ in $T\cap\Omega^\varepsilon$ for $T\in\mathcal{T}_H$. We can thus eliminate the pressure from (3.217)–(3.218) by introducing a subspace of $\widehat{V}_H^{Os}$:

$$\widehat{Z}_H^{Os}=\left\{\boldsymbol{v}\in\widehat{V}_H^{Os}\text{ such that }\operatorname{div}\boldsymbol{v}=0, \ \forall T\in\mathcal{T}_H\right\}$$

Thus system (3.217)–(3.218) is equivalent to: find $\boldsymbol{u}_H \in \widehat{Z}_H^{Os}$ such that

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\left(\mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} + \rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{u}_H\cdot\boldsymbol{v}\right) = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\boldsymbol{f}\cdot\boldsymbol{v},\ \forall\boldsymbol{v}\in\widehat{Z}_H^{Os}$$

The existence and uniqueness of a velocity $\boldsymbol{u}_H$ to this problem is guaranteed by Theorem 2.1.2. Then the existence and uniqueness of a pressure $\bar{p}_H$ follows from the fact that div $\widehat{V}_H^{Os} = \widehat{M}_H$. We conclude that (3.217)–(3.218) has one and only one solution $(\boldsymbol{u}_H, \bar{p}_H) \in \widehat{V}_H^{Os} \times \widehat{M}_H$.

Moreover, like other multiscale methods previously presented in this chapter, this high-order multiscale method can also be applied to solve Navier-Stokes problems on the coarse mesh. The reader can refer to subsection 3.3.5 for more details.

3.7.5 The reconstruction of fine-scale features

After solving the coarse-scale problems, we obtain the coarse-scale solutions $\boldsymbol{u}_H$ and $\bar{p}_H$. Then we reconstruct the fine-scale features of the solution for visualization. On each coarse element $T \in \mathcal{T}_H$, making use of $\widehat{V}_H^{Os}$ defined by (3.215) and the formula of π_H defined by (3.216), the fine-scale velocity and pressure are reconstructed by

$$\boldsymbol{u}_H|_T = \sum_{E\in\mathcal{E}(T)}\sum_{i=1}^{s} u_{E,i}\boldsymbol{\Phi}_{E,i} + \sum_{k=1}^{r} u_{T,k}\boldsymbol{\Psi}_{T,k}$$

$$p_H|_T = \sum_{E\in\mathcal{E}(T)}\sum_{i=1}^{s} u_{E,i}\pi_{E,i} + \sum_{k=1}^{r} u_{T,k}\pi_{T,k} + \bar{p}_H|_T$$

Chapter 4

Technical aspects of Crouzeix-Raviart multiscale finite element methods

This chapter addresses some technical aspects of the implementation of Crouzeix-Raviart multiscale finite element methods in *TrioCFD*. Local problems are discretized by the finite volume element method where physical unknowns are discretized by the Crouzeix-Raviart finite element. In particular, in high-order Crouzeix-Raviart multiscale finite element methods, we also discretize the pressure of local problems in the $\mathbb{P}_1$ space instead of the $\mathbb{P}_0$ space. The idea is to improve the accuracy of the fine-scale pressure given by the high-order multiscale method.

Coarse-scale problems are discretized by a Galerkin method where basis functions are solutions of local problems. In multiscale finite element methods, coarse-scale problems are defined on the coarse mesh by assembling matrices pre-computed locally on coarse elements. When computing matrices on each coarse element, we decompose multiscale basis functions using the Crouzeix-Raviart finite element basis functions. The properties of the Crouzeix-Raviart finite element basis functions facilitate the computation of integrals in the coefficients of matrices.

In an early stage of this thesis, we solved both local and coarse-scale problems by adapting the *prediction-correction* algorithm [42, 135] which is the only solver natively available in *TrioCFD*. However, we noticed some limitations of this algorithm, especially when solving local and coarse-scale problems defined in high-order multiscale methods. Therefore in a later stage of this thesis, an important work was devoted to the implementation of a direct solver to solve local and coarse-scale problems efficiently.

Outline Section 4.1 presents the finite volume element method and the Crouzeix-Raviart element. Section 4.2 and section 4.3 present respectively the discretization and the solution of local problems with the *prediction-correction* algorithm. Section 4.4 and section 4.5 reveal respectively the discretization and the solution of coarse-scale problems with the *prediction-correction* algorithm. Section 4.6 presents the discretization of local and coarse-scale problems defined in high-order multiscale methods and a direct solver.

4.1 The finite volume element method

In this thesis, we implement Crouzeix-Raviart multiscale finite element methods in *TrioCFD* which focus on the finite volume element method. This software will be presented in Chapter 5. As indicated by its name, the finite volume element method can be considered as a certain combination of the finite element method (see Chapter 2) and the finite volume method [141].

The finite volume method is a widely used discretization technique for partial differential equations. One reason of its popularity is that the method ensures a local mass conservation, which is important for some physical problems. The finite volume method is based on a balance approach: the global domain is discretized into a series of control volumes and a local balance is performed on each control volume. By the divergence theorem, we obtain an integral formulation of the fluxes on the boundary of the control volume. Then the integral formulation will be discretized with respect to discrete unknowns. However, the finite volume method requires a scheme for approximating certain fluxes.

The finite volume element method [34] was developed as an attempt to use the ideas of the finite element method to create a more systematic finite volume methodology. The main idea of the finite volume element method is to approximate the fluxes on the boundary of control volumes by replacing unknowns by a finite element approximation. The finite volume element method consists of two meshes: a primal mesh and a dual mesh. The primal mesh is used to approximate the unknowns whereas the dual mesh is used to discretize equations. In other words, equations are discretized by volume integrals on control volumes of the dual mesh. Then physical unknowns are discretized into a finite element space on the primal mesh.

The finite volume element method has the following advantages: (1) the accuracy of the method can be improved by using more accurate finite element spaces; (2) the method offers more flexibility to handle complicated geometries and simplify the treatment of boundary conditions; (3) the method can be viewed as a perturbation to the finite element method and can use existing theories of the finite element method. The reader can refer to [109] and the references therein for a detailed review of the finite volume element method.

In *TrioCFD*, the finite volume element method was first developed for the Crouzeix-Raviart finite element in [65]. The convergence of the method was proved for the Stokes equations by using results and techniques of the finite element method. Then the finite volume element method has been further developed in [82, 123, 136].

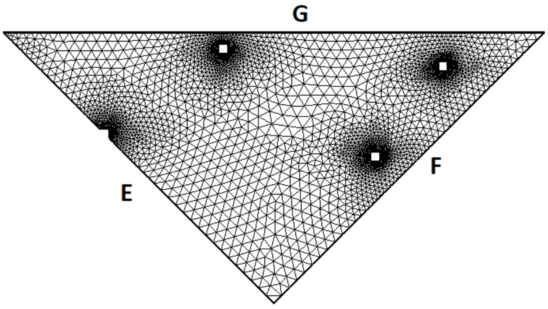


FIGURE 4.1: The fine mesh on a coarse element perforated by obstacles

4.1.1 The Crouzeix-Raviart finite element and the control volume

For each coarse element $T \in \mathcal{T}_H$, we denote the fine mesh on this coarse element by $\mathcal{T}_h(T)$. In the fine mesh $\mathcal{T}_h(T)$, we denote the number of elements by N_k and the number of faces by N_f. The mesh $\mathcal{T}_h(T)$ is the primal mesh and then we associate to this mesh a dual mesh consisting of control volumes. In this chapter, we present only the implementation of Crouzeix-Raviart multiscale finite element methods in two dimensions.

In this thesis, we define a new boundary condition named *averaged* Dirichlet boundary condition to stand for the integral type boundary condition. As shown in Figure 4.1, the boundary of $T \cap \Omega^\varepsilon$ is decomposed into two parts:

$$\partial(T \cap \Omega^\varepsilon) = \Gamma_d \cup \Gamma_a$$

where Γ_d is the boundary of obstacles and Γ_a defined by

$$\Gamma_a = \mathcal{E}(T) \cap \Omega^\varepsilon \text{ with } \mathcal{E}(T) = E \cup F \cup G$$

Note that $\mathcal{E}(T)$ is the set of boundaries of T (not cut by obstacles). We impose the no-slip Dirichlet condition on Γ_d and the *averaged* Dirichlet condition on Γ_a.

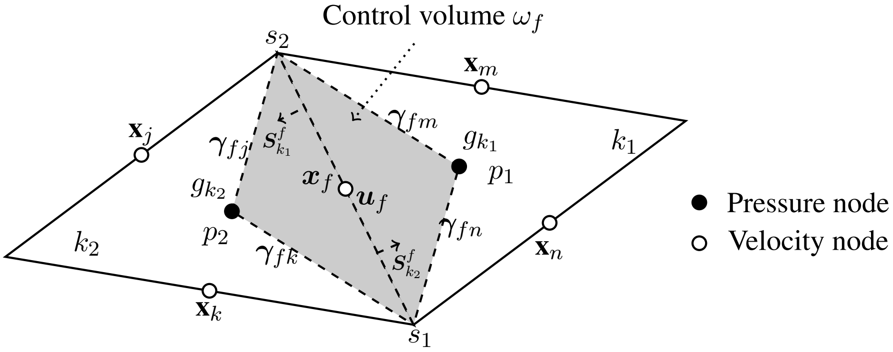


FIGURE 4.2: The Crouzeix-Raviart element and control volumes

Figure 4.2 illustrates a two dimensional Crouzeix-Raviart element which was already presented in section 2.4. The pressure unknown is located at the barycenter of the element whereas the velocity unknown is located at the barycenter of faces. For a velocity node $\boldsymbol{x}_f$ located in the interior of $T \cap \Omega^\varepsilon$, its control volume is denoted by ω_f and is constructed by joining vertices of the face f with barycenters g_{k_1} and g_{k_2} of elements k_1 and k_2. The control volume of the pressure node is the element itself. The boundary of the control volume ω_f is denoted by $\partial\omega_f$. As shown in Figure 4.2, denoting the facet between the node f and the node j by γ_{fj}, then $\partial\omega_f$ is decomposed as

$$\partial\omega_f = \gamma_{fj} \cup \gamma_{fm} \cup \gamma_{fn} \cup \gamma_{fk} \tag{4.1}$$

Definition 4.1.1. *For any element $k \in \mathcal{T}_h(T)$, for any velocity node $\boldsymbol{x}_f$ in k, we denote the unit outward vector normal to face f by $\boldsymbol{n}^f_k$ and the measure of the face by* $\operatorname{meas}(f)$*. Then we define **the outward normal surface vector** to the face f by*

$$\boldsymbol{S}^f_k = \operatorname{meas}(f) \cdot \boldsymbol{n}^f_k \tag{4.2}$$

As shown in Figure 4.2, for the node $\boldsymbol{x}_f$, the outward normal surface vector to the face f is $\boldsymbol{S}^f_{k_1}$ in k_1 and $\boldsymbol{S}^f_{k_2}$ in k_2 with $\boldsymbol{S}^f_{k_1} + \boldsymbol{S}^f_{k_2} = \mathbf{0}$.

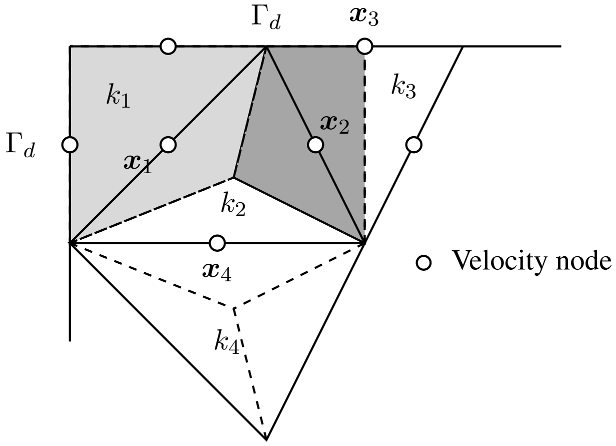


FIGURE 4.3: The treatment of control volumes with respect to Dirichlet nodes

Control volumes of nodes located on the boundary Γ_d In fact, in the finite volume element method, the conservation of momentum equation (see subsection 4.2.2) is not performed on control volumes of Dirichlet nodes, i.e. nodes on Γ_d. In order to ensure the conservation of momentum equation in the entire computational domain, a special treatment is necessary for control volumes of Dirichlet nodes. As shown in Figure 4.3, we modify the control volume of velocity according to the number of Dirichlet nodes in one element. This modification was originally presented in [65].

- if the element has no Dirichlet node, for example, in the element k_4, the control volume of the internal node $\boldsymbol{x}_4$ is constructed by joining vertices with the barycenter of the element. As a result, the total control volume of this node is $\frac{1}{3}\text{meas}\,(k_2) + \frac{1}{3}\text{meas}\,(k_4)$.
- if the element has only one Dirichlet node, for example, in the element k_3, the control volume of node $\boldsymbol{x}_2$ in k_3 is constructed by joining the Dirichlet node $\boldsymbol{x}_3$ with the vertex faced to it. As a result, the total control volume of the node $\boldsymbol{x}_2$ is $\frac{1}{3}\text{meas}\,(k_2)+\frac{1}{2}\text{meas}\,(k_3)$.
- if the element has two Dirichlet nodes, for example, in the element k_1, the control volume of node $\boldsymbol{x}_1$ in k_1 is the entire element. As a result, the total control volume of the node $\boldsymbol{x}_1$ is $\text{meas}\,(k_1) + \frac{1}{3}\text{meas}\,(k_2)$

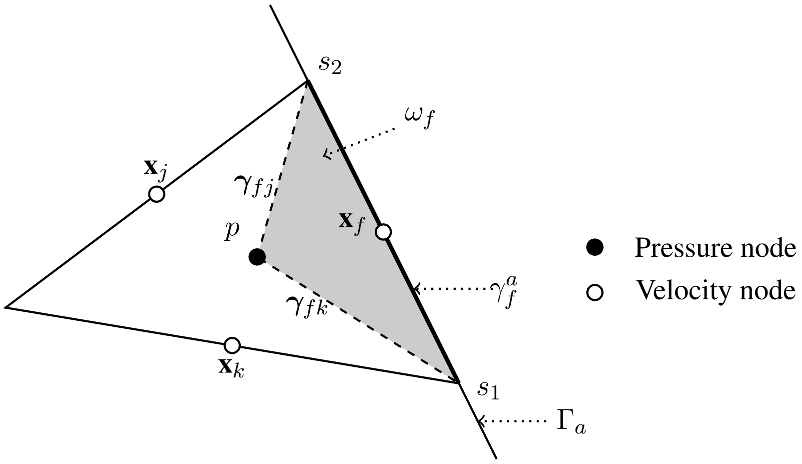


FIGURE 4.4: The control volume of a velocity node on the boundary Γ_a

Control volumes of nodes located on the boundary Γ_a For a velocity node $\boldsymbol{x}_f$ located on the boundary Γ_a, we decompose the boundary of its control volume $\partial\omega_f$ into two parts:

$$\gamma_f^a = \partial\omega_f \cap \Gamma_a, \quad \gamma_f = \partial\omega_f \setminus \gamma_f^a \tag{4.3}$$

For the node $\boldsymbol{x}_f$ shown in Figure 4.4, we have $\gamma_f^a = [s_1 s_2]$ and $\gamma_f = \gamma_{fj} \cup \gamma_{fk}$. In fact, for an internal node, $\gamma_f^a = \varnothing$ and $\gamma_f = \partial\omega_f$. Using this convention, for any node located in the interior of the domain or on the boundary Γ_a, the boundary of the control volume ω_f is always decomposed into γ_f^a and γ_f.

For the coarse element $T \in \mathcal{T}_H$, we recall the finite element space of velocity $V_h(T)$ and pressure $M_h(T)$ defined for the Crouzeix-Raviart finite element in section 2.4:

$$V_h\,(T) = \left\{ \boldsymbol{v}_h \in \left(L^2\,(T \cap \Omega^\varepsilon)\right)^d \;\middle|\; \boldsymbol{v}_h|_k \in (\mathbb{P}_1(k))^d\,,\ \boldsymbol{v}_h \text{ continuous at } \boldsymbol{x}_i,\ \forall k \in T_h\,(T) \right\} \tag{4.4}$$

$$M_h\,(T) = \left\{ q_h \in L^2\,(T \cap \Omega^\varepsilon) \;\middle|\; q_h|_k \in \mathbb{P}_0\,(k)\,,\ \forall k \in T_h\,(T) \right\}$$

Let $\{\phi_1, \cdots, \phi_{N_f}\}$ be a basis of each component of $V_h(T)$ where $\phi_i(x_j) = \delta_{ij}$ for $i, j = 1, \cdots, N_f$. Let $\{\psi_1, \cdots, \psi_{N_k}\}$ be a basis of $M_h(T)$ where ψ_k is the characteristic function of the element k for $k \in \mathcal{T}_h(T)$. We assume that velocity nodes in $\mathcal{T}_h(T)$ are numbered in a way that the first N_{f0} are non Dirichlet nodes, in other words,

$$\begin{cases} 1 \leq f \leq N_{f0} \Rightarrow x_f \notin \Gamma_d \\ N_{f0} < f \leq N_f \Rightarrow x_f \in \Gamma_d \end{cases} \tag{4.5}$$

Note that non Dirichlet nodes include nodes in the interior of $T \cap \Omega^\varepsilon$ and on the boundary Γ_a.

4.2 The discretization of local problems

In [95, 117, 118], local problems are discretized by the finite element method and the penalization method is applied to solve local problems on a simple uniform Cartesien grid. It should be noted that when using the penalization method, it is required to couple the penalization parameter to the mesh size and the discrete system may be ill-conditioned if the penalization parameter is too small.

In *TrioCFD*, no penalization method is used. The body-fitted unstructured meshes are used to resolve the boundary of obstacles and no-slip condition is imposed directly on obstacles. In this section, we discretize the local problem defined by Stokes equations (3.39) with the finite volume element method and the Crouzeix-Raviart element. The discretization of the Oseen term in the local problem (3.102) will be discussed in subsection 4.2.3.

On any $T \in \mathcal{T}_H$, for each $E \in \mathcal{E}(T)$ and for $i = 1, \cdots, s$, the local problem (3.39) is defined as: find $\mathbf{\Phi}_{E,i} \in V_h(T)$, $\pi_{E,i} \in M_h(T)$ and $\alpha_{E,i} \in \mathbb{R}$ such that

$$-\mu\Delta\mathbf{\Phi}_{E,i} + \nabla\pi_{E,i} = \mathbf{0} \text{ on } T \cap \Omega^\varepsilon, \tag{4.6}$$

$$\operatorname{div} \mathbf{\Phi}_{E,i} = \alpha_{E,i} \text{ on } T \cap \Omega^\varepsilon, \tag{4.7}$$

$$\mu\nabla\mathbf{\Phi}_{E,i}\boldsymbol{n} - \pi_{E,i}\boldsymbol{n} \in \operatorname{span}\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\} \text{ on } F \cap \Omega^\varepsilon,\ \forall F \in \mathcal{E}(T), \tag{4.8}$$

$$\mathbf{\Phi}_{E,i} = \mathbf{0} \text{ on } \partial B^\varepsilon \cap T, \tag{4.9}$$

$$\int_F \mathbf{\Phi}_{E,i} \cdot \boldsymbol{\omega}_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases} \quad \forall F \in \mathcal{E}(T),\ j = 1, \cdots, s. \tag{4.10}$$

$$\int_{T\cap\Omega^\varepsilon} \pi_{E,i} = 0. \tag{4.11}$$

The constant $\alpha_{E,i}$ is computed by the relation $\int_{T\cap\Omega^\varepsilon} \alpha_{E,i} = \int_{\partial(T\cap\Omega^\varepsilon)} \mathbf{\Phi}_{E,i} \cdot \boldsymbol{n}$. We assume that μ is constant in the entire domain $T \cap \Omega^\varepsilon$.

The velocity and pressure unknowns can be represented using basis of $V_h(T)$ and $M_h(T)$ as

$$\forall E \in \mathcal{E}(T),\ i = 1, \cdots, s,\ \mathbf{\Phi}_{E,i} = \sum_{g=1}^{N_f} \mathbf{\Phi}_{E,i}^g \phi_g, \quad \pi_{E,i} = \sum_{k=1}^{N_k} \pi_{E,i}^k \psi_k \tag{4.12}$$

Note that throughout this chapter, we will abuse the index by using k also as the index of the element k. We use f to represent the index of the face f. We use $\mathbf{U}_h$ and $\mathbf{P}_h$ to denote respectively the vector containing the discretized velocity and pressure:

$$\mathbf{U}_h = \left(\mathbf{\Phi}_{E,i}^1, \cdots, \mathbf{\Phi}_{E,i}^{N_f}\right), \quad \mathbf{P}_h = \left(\pi_{E,i}^1, \cdots, \pi_{E,i}^{N_k}\right)$$

4.2.1 The conservation of mass

For any element $k \in \mathcal{T}_h(T)$, we integrate equation (4.7) over k:

$$\int_k \text{div } \mathbf{\Phi}_{E,i} = \int_k \alpha_{E,i}$$

Denoting by $\boldsymbol{n}$ the unit outward vector normal to ∂k and applying the divergence theorem, we obtain

$$\int_{\partial k \backslash \Gamma_d} \mathbf{\Phi}_{E,i} \cdot \boldsymbol{n} = \alpha_{E,i} \cdot \text{meas}(k)$$

where $\text{meas}(k)$ is the measure of k. We recall that $\mathbf{\Phi}_{E,i} = \mathbf{0}$ on Γ_d by equation (4.9).

Substituting $\mathbf{\Phi}_{E,i}$ defined by (4.12) into this equation, we obtain

$$\sum_{g=1}^{N_{f0}} \mathbf{\Phi}_{E,i}^g \int_{\partial k \backslash \Gamma_d} \phi_g \boldsymbol{n} = \alpha_{E,i} \cdot \text{meas}(k)$$

This system can be written in the matrix form:

$$\mathbf{B}_h \mathbf{U}_h = \mathbf{G}_h$$

where

$$\forall k \in \mathcal{T}_h(T),\ g \leq N_{f0},\ (\mathbf{B}_h)_{k,g} = -\int_{\partial k \backslash \Gamma_d} \phi_g \boldsymbol{n} = -\boldsymbol{S}_k^g$$
$$\forall k \in \mathcal{T}_h(T),\ (\mathbf{G}_h)_k = -\alpha_{E,i} \cdot \text{meas}(k)$$

We recall that the vector $\boldsymbol{S}_k^g$ is defined by (4.2).

4.2.2 The conservation of momentum equation

First of all, we recall that $\mathbf{\Phi}_{E,i} = \mathbf{0}$ on Γ_d, i.e. for Dirichlet nodes $N_{f0} < f \leq N_f$. For non Dirichlet nodes $f = 1, \cdots, N_{f0}$ (including internal nodes and nodes on Γ_a), integrating equation (4.6) over ω_f (see Figure 4.2 and Figure 4.4) and applying the divergence theorem, we obtain

$$\int_{\gamma_f} (-\mu \nabla \mathbf{\Phi}_{E,i} \boldsymbol{n} + \pi_{E,i} \boldsymbol{n}) = \int_{\gamma_f^a} (\mu \nabla \mathbf{\Phi}_{E,i} \boldsymbol{n} - \pi_{E,i} \boldsymbol{n}) \tag{4.13}$$

Formula (4.3) shows that $\gamma_f^a = \partial \boldsymbol{\omega}_f \cap \Gamma_a$ with $\Gamma_a = \mathcal{E}(T) \cap \Omega^\varepsilon$.

Equation (4.8) implies that there exist $\lambda_{F,1}, \cdots, \lambda_{F,s} \in \mathbb{R}$ such that

$$\mu \nabla \mathbf{\Phi}_{E,i} \boldsymbol{n} - \pi_{E,i} \boldsymbol{n} = \sum_{j=1}^{s} \lambda_{F,j} \boldsymbol{\omega}_{F,j} \text{ on } F \cap \Omega^\varepsilon, \text{ for all } F \in \mathcal{E}(T)$$

Therefore (4.13) can be written as

$$\int_{\gamma_f} (-\mu \nabla \mathbf{\Phi}_{E,i} \boldsymbol{n} + \pi_{E,i} \boldsymbol{n}) = \sum_{F \in \mathcal{E}(T)} \sum_{j=1}^{s} \lambda_{F,j} \int_{\partial \omega_f \cap F} \boldsymbol{\omega}_{F,j}$$

Substituting $\mathbf{\Phi}_{E,i}$ and $\pi_{E,i}$ defined by (4.12) into this equation, we obtain

$$\sum_{g=1}^{N_{f0}} \left(-\mathbf{\Phi}_{E,i}^{g} \int_{\gamma_f} \mu \nabla \phi_g \cdot \boldsymbol{n} \right) + \sum_{k=1}^{N_k} \left(\pi_{E,i}^{k} \int_{\gamma_f} \psi_k \boldsymbol{n} \right) = \sum_{F \in \mathcal{E}(T)} \sum_{j=1}^{s} \lambda_{F,j} \int_{\partial\omega_f \cap F} \boldsymbol{\omega}_{F,j}$$

Denoting $\mathbf{\Lambda}_h = (\lambda_{F,1}, \cdots, \lambda_{F,s})_{F \in \mathcal{E}(T)}$, this system can be written in the matrix form:

$$\mathbf{A}_h \mathbf{U}_h + \mathbf{C}_h \mathbf{P}_h + \mathbf{E}_h \mathbf{\Lambda}_h = \mathbf{0} \tag{4.14}$$

where

- $\mathbf{A}_h$ is a matrix which contains two blocks A_h corresponding to each component of velocity in two dimensions.

  $$\mathbf{A}_h = \begin{pmatrix} A_h & 0 \\ 0 & A_h \end{pmatrix}$$

  where

  $$\forall f,\ g \le N_{f0},\ (A_h)_{f,g} = -\int_{\gamma_f} \mu \nabla \phi_g \cdot \boldsymbol{n}$$

  We denote the element containing two different faces f and g by k_{fg} and denote its measure by $\text{meas}\,(k_{fg})$. The basis functions ϕ_f and ϕ_g are linear inside the element k_{fg} and their gradients are constant inside this element. It was proved in [65] that

  $$\nabla \phi_g|_{k_{fg}} = \frac{1}{\text{meas}\,(k_{fg})} \int_{\partial k_{fg}} \phi_g \boldsymbol{n} = \frac{\boldsymbol{S}_{k_{fg}}^{g}}{\text{meas}\,(k_{fg})} \tag{4.15}$$

  Thus the coefficient $(A_h)_{f,g}$ is

  $$(A_h)_{f,g} = \nabla \phi_g|_{k_{fg}} \cdot \int_{\partial\omega_f \cap k_{fg}} \mu \boldsymbol{n} = \frac{\boldsymbol{S}_{k_{fg}}^{g}}{\text{meas}\,(k_{fg})} \cdot \int_{\partial\omega_f \cap k_{fg}} \mu \boldsymbol{n}$$

  Since μ is constant in k_{fg}, it is easy to check that

  $$\int_{\partial\omega_f \cap k_{fg}} \mu \boldsymbol{n} = -\mu \boldsymbol{S}_{k_{fg}}^{f}$$

  Finally, the coefficient $(A_h)_{f,g}$ is

  $$(A_h)_{f,g} = \mu \frac{\boldsymbol{S}_{k_{fg}}^{f} \cdot \boldsymbol{S}_{k_{fg}}^{g}}{\text{meas}\,(k_{fg})}$$

- $\mathbf{C}_h$ is a matrix with

  $$\forall f \le N_{f0},\ \ k \le N_k,\ (\mathbf{C}_h)_{f,k} = \int_{\gamma_f} \psi_k \boldsymbol{n}$$

  It was proved in [65] that

  $$(\mathbf{C}_h)_{f,k} = -\boldsymbol{S}_k^{f}$$

We recall that $\boldsymbol{S}_k^f$ is defined by (4.2). By comparing the coefficients of $\mathbf{B}_h$ and $\mathbf{C}_h$, it is easy to verify that $\mathbf{C}_h = \mathbf{B}_h^t$.

- $\mathbf{E}_h$ is a matrix with

$$\forall f \leq N_{f0},\ F \in \mathcal{E}(T),\ j = 1, \cdots, s,\ (\mathbf{E}_h)_{f,(F,j)} = \int_{\partial\omega_f \cap F} \boldsymbol{\omega}_{F,j} \tag{4.16}$$

In two dimensions, $\mathcal{E}(T)$ contains three boundaries. Thus $\mathbf{E}_h$ is a matrix with $N_f \times d$ rows and $3 \times s$ columns.

4.2.3 The discretization of the Oseen term

Now we recall the local problems defined by Oseen equations (3.102). The Oseen velocity $\boldsymbol{U}_o$ is assumed to be a constant vector field in the entire domain $T \cap \Omega^\varepsilon$ and thus div $\boldsymbol{U}_o = 0$. On any $T \in \mathcal{T}_H$, for each $E \in \mathcal{E}(T)$ and for $i = 1, \cdots, s$, the Oseen local problem is defined by: find $\boldsymbol{\Phi}_{E,i} \in V_h(T)$, $\pi_{E,i} \in M_h(T)$ and $\alpha_{E,i} \in \mathbb{R}$ such that

$$-\mu\Delta\boldsymbol{\Phi}_{E,i} + \rho\,(\boldsymbol{U}_o \cdot \nabla)\,\boldsymbol{\Phi}_{E,i} + \nabla\pi_{E,i} = \mathbf{0} \text{ in } T \cap \Omega^\varepsilon, \tag{4.17}$$

$$\text{div } \boldsymbol{\Phi}_{E,i} = \alpha_{E,i} \text{ in } T \cap \Omega^\varepsilon,$$

$$\mu\nabla\boldsymbol{\Phi}_{E,i}\boldsymbol{n} - \frac{1}{2}\rho\,(\boldsymbol{U}_o \cdot \boldsymbol{n})\,\boldsymbol{\Phi}_{E,i} - \pi_{E,i}\boldsymbol{n} \in \text{span}\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\} \text{ on } F \cap \Omega^\varepsilon,\ \forall F \in \mathcal{E}(T), \tag{4.18}$$

$$\boldsymbol{\Phi}_{E,i} = \mathbf{0} \text{ on } \partial B^\varepsilon \cap T$$

$$\int_{F\cap\Omega^\varepsilon} \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{\omega}_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases} \quad \forall F \in \mathcal{E}\,(T)\ \ j = 1, \cdots, s.$$

$$\int_{T\cap\Omega^\varepsilon} \pi_{E,i} = 0.$$

where $\mathcal{E}(T)$ is the set of faces of T. The constant $\alpha_{E,i}$ is computed by the relation $\int_{T\cap\Omega^\varepsilon} \alpha_{E,i} = \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{n}$. Throughout this section, we assume that both μ and ρ are constant on $T \cap \Omega^\varepsilon$.

For any velocity node $\boldsymbol{x}_f$ in the domain, integrating the momentum equation (4.17) over its control volume ω_f gives

$$\int_{\omega_f} \left(-\mu\nabla\boldsymbol{\Phi}_{E,i}\boldsymbol{n} + \rho\,(\boldsymbol{U}_o \cdot \nabla)\,\boldsymbol{\Phi}_{E,i} + \pi_{E,i}\boldsymbol{n}\right) = \int_{\partial\omega_f} \left(\mu\nabla\boldsymbol{\Phi}_{E,i}\boldsymbol{n} - \rho\,(\boldsymbol{U}_o \cdot \boldsymbol{n})\,\boldsymbol{\Phi}_{E,i} - \pi_{E,i}\boldsymbol{n}\right) \tag{4.19}$$

Compared to equation (4.13), there is an additional Oseen term in the equation above. Thus we present specially the discretization of the Oseen term in what follows. We distinguish two types of nodes: nodes in the interior of the domain and nodes on the boundary Γ_a.

Nodes in the interior of the domain For a velocity node $\boldsymbol{x}_f$ located in the interior of the domain, we denote by $\boldsymbol{n}_f$ the unit outward vector normal to $\partial\omega_f$. As shown in Figure 4.5, denoting the facet between the node f and the node j by γ_{fj}, we recall that the boundary of the control volume ω_f is decomposed as

$$\partial\omega_f = \gamma_{fj} \cup \gamma_{fm} \cup \gamma_{fn} \cup \gamma_{fk}$$

Now we present the integral of the Oseen term in (4.19) over the facet γ_{fj}. The integrals over other facets are computed in the same way. In *TrioCFD*, the Upwind scheme and other schemes are used to discretize the nonlinear convection term in Navier-Stokes equations. In this thesis, we

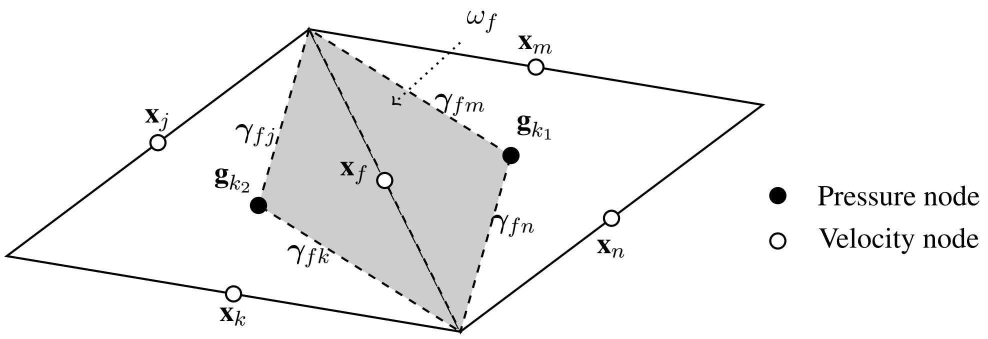


FIGURE 4.5: The control volume of an internal node and its boundary $\partial\omega_f$

adapt this scheme to discretize the Oseen term in local Oseen problems. The idea of the Upwind scheme [73, 125] is to approximate $\mathbf{\Phi}_{E,i}$ on γ_{fj} by

$$\begin{cases} \mathbf{\Phi}_{E,i} = \mathbf{\Phi}^f_{E,i} & \text{if } (\boldsymbol{U}_0 \cdot \boldsymbol{n}_f) > 0 \\ \mathbf{\Phi}_{E,i} = \mathbf{\Phi}^j_{E,i} & \text{if } (\boldsymbol{U}_0 \cdot \boldsymbol{n}_j) > 0 \end{cases}$$

where $\boldsymbol{n}_f = -\boldsymbol{n}_j$. In other words, the integral over γ_{fj} of the Oseen term is approximated by

$$\int_{\gamma_{fj}} \rho \mathbf{\Phi}_{E,i} \left(\boldsymbol{U}_o \cdot \boldsymbol{n}_f\right) \approx \int_{\gamma_{fj}} \rho \mathbf{\Phi}^f_{E,i} \max\left(\boldsymbol{U}_o \cdot \boldsymbol{n}_f, 0\right) + \int_{\gamma_{fj}} \rho \mathbf{\Phi}^j_{E,i} \max\left(\boldsymbol{U}_o \cdot \boldsymbol{n}_j, 0\right) \qquad (4.20)$$

We denote by $\mathbf{L}_h\left(\boldsymbol{U}_o\right)$ the matrix associated to the Oseen term:

$$\mathbf{L}_h\left(\boldsymbol{U}_o\right) = \begin{pmatrix} L_h\left(\boldsymbol{U}_o\right) & 0 \\ 0 & L_h\left(\boldsymbol{U}_o\right) \end{pmatrix}$$

Note that this is a block matrix with $L_h\left(\boldsymbol{U}_o\right)$ corresponding to each component of velocity in two dimensions. The diagonal and non-diagonal coefficients of $L_h\left(\boldsymbol{U}_o\right)$ are

$$\forall f \leq N_{f0},\ L_h\left(\boldsymbol{U}_o\right)_{f,f} = \int_{\partial\omega_f} \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \boldsymbol{n}_f + \left|\boldsymbol{U}_o \cdot \boldsymbol{n}_f\right|\right)$$
$$\forall g \leq N_{f0},\ f \neq g,\ L_h\left(\boldsymbol{U}_o\right)_{f,g} = \int_{\gamma_{fg}} \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \boldsymbol{n}_f - \left|\boldsymbol{U}_o \cdot \boldsymbol{n}_f\right|\right)$$

Nodes on the boundary Γ_a We present now the treatment of nodes located on the boundary Γ_a where we impose the integral type boundary condition. As shown in Figure 4.6, for any node $\boldsymbol{x}_f$ located on Γ_a, we decompose the boundary of the control volume ω_f as

$$\partial\omega_f = \gamma_{fj} \cup \gamma_{fk} \cup \gamma^a_f$$

Thus the integral in the right hand side of (4.19) can be decomposed into three integrals γ_{fj}, γ_{fk} and γ^a_f. Since γ_{fj} and γ_{fk} are in the interior of the domain, we can discretize integrals on γ_{fj} and γ_{fk} in the same way as in (4.20). Thus the non-diagonal coefficient of $L_h\left(\boldsymbol{U}_o\right)$ for any node $\boldsymbol{x}_f$ located on Γ_a is

$$\forall g \leq N_{f0},\ f \neq g,\ L_h\left(\boldsymbol{U}_o\right)_{f,g} = \int_{\gamma_{fg}} \frac{1}{2}\rho\left(\boldsymbol{U}_o \cdot \boldsymbol{n}_f - \left|\boldsymbol{U}_o \cdot \boldsymbol{n}_f\right|\right)$$

We present the discretization of the integral on γ^a_f in what follows.

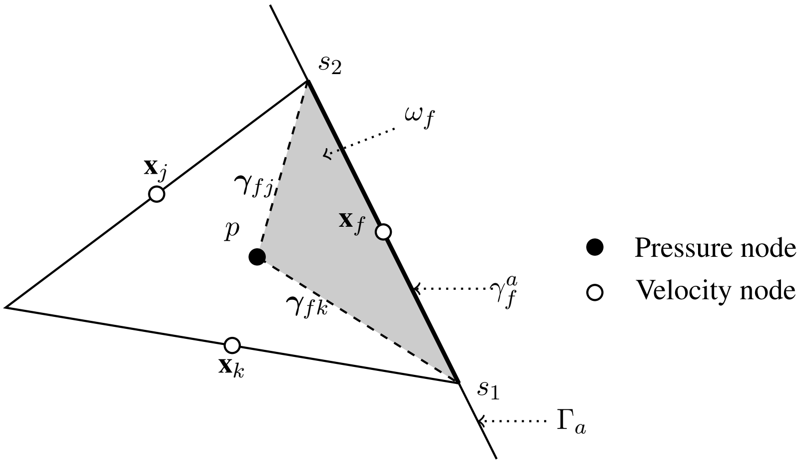


FIGURE 4.6: The control volume of a velocity node on the boundary Γ_a

Equation (4.18) implies that there exist $\lambda_{F,1}, \cdots, \lambda_{F,s} \in \mathbb{R}$ such that

$$\mu \nabla \mathbf{\Phi}_{E,i} \boldsymbol{n} - \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}\right) \mathbf{\Phi}_{E,i} - \pi_{E,i}\boldsymbol{n} = \sum_{j=1}^{s} \lambda_{F,j} \boldsymbol{\omega}_{F,j} \text{ on } F \cap \Omega^{\varepsilon}, \ \forall F \in \mathcal{E}(T)$$

Making use of this equality, we can write

$$\begin{aligned}\int_{\gamma_f^a} \left(\mu \nabla \mathbf{\Phi}_{E,i} \boldsymbol{n} - \rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}\right) \mathbf{\Phi}_{E,i} - \pi_{E,i}\boldsymbol{n}\right) = & \sum_{F \in \mathcal{E}(T)} \sum_{j=1}^{s} \lambda_{F,j} \int_{\gamma_f^a \cap F} \boldsymbol{\omega}_{F,j} \\ & - \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}_f\right) \int_{\gamma_f^a} \mathbf{\Phi}_{E,i}\end{aligned}$$

It is easy to check that $\int_{\gamma_f^a} \phi_g = \text{meas}(\gamma_f^a)$. Making use of $\mathbf{\Phi}_{E,i}$ defined by (4.12), we have

$$\frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}_f\right) \int_{\gamma_f^a} \mathbf{\Phi}_{E,i} = \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}_f\right) \text{meas}(\gamma_f^a) \mathbf{\Phi}_{E,i}^{f}$$

Consequently, the diagonal coefficient of $L_h\left(\boldsymbol{U}_o\right)$ for any node located on Γ_a is

$$L_h\left(\boldsymbol{U}_o\right)_{f,f} = \frac{1}{2}\rho \left(\boldsymbol{U}_o \cdot \boldsymbol{n}_f\right) \text{meas}(\gamma_f^a), \ \forall f \in \Gamma_a$$

4.2.4 The discretization of the velocity integral boundary condition

Now let us discretize (4.10) for $F \in \mathcal{E}\left(T\right)$ and $j = 1, \cdots, s$.

$$\sum_{f=1}^{N_{f0}} \int_{F \cap \partial \omega_f} \mathbf{\Phi}_{E,i} \cdot \boldsymbol{\omega}_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases}$$

It is easy to check that this system can be written in the matrix form:

$$\mathbf{D}_h \mathbf{U}_h = \mathbf{H}_h$$

where

$$\forall F \in \mathcal{E}(T),\ j = 1, \cdots, s,\ f \leq N_{f0},\ (\mathbf{D}_h)_{(F,j),f} = \int_{F \cap \partial\omega_f} \boldsymbol{\omega}_{F,j}$$

$$\forall F \in \mathcal{E}(T),\ j = 1, \cdots, s,\ (\mathbf{H}_h)_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases}$$

It is easy to see that $(\mathbf{D}_h)$ is a matrix with $3 \times s$ rows and $N_f \times d$ columns. Comparing the coefficients of $\mathbf{D}_h$ and $\mathbf{E}_h$ defined by (4.16), we conclude that $\mathbf{E}_h = \mathbf{D}_h^t$.

Until now, we have not presented the discretization of equation (4.11). In *TrioCFD*, when solving a flow problem with the *prediction-correction* algorithm (see section 4.3), the average of pressure is not imposed to be zero in the computational domain, i.e. equation (4.11) is not satisfied. The uniqueness of pressure is guaranteed by taking the pressure imposed on the boundary (by the user in the input file) as a reference. In this thesis, in order to obtain a pressure which satisfies (4.11), we first compute the volume average of the pressure solution given by *TrioCFD* and then subtract the average from the pressure solution. As a result, (4.11) is satisfied.

4.2.5 The discretization of the temporal inertial term

In *TrioCFD*, the only solver natively available is the *prediction-correction* algorithm (see section 4.3) which is a time-marching technique for the unsteady Navier-Stokes equations. This algorithm can also solve steady problems but it still passes through a pseudo-transient process. In the first stage of this thesis, we have adapted this algorithm to solve steady local problems. As we are going to present in section 4.6, we have implemented later in this thesis a direct solver to solve steady local and coarse scale problems without passing through the transient process.

In the *prediction-correction* algorithm, the unsteady $\boldsymbol{\Phi}_{E,i}$ and $\pi_{E,i}$ can be written as:

$$\boldsymbol{\Phi}_{E,i}(t) = \sum_{g=1}^{N_f} \boldsymbol{\Phi}_{E,i}^g(t)\,\phi_g, \quad \pi_{E,i}(t) = \sum_{k=1}^{N_k} \pi_{E,i}^k(t)\,\psi_k$$

For $f = 1, \cdots, N_{f0}$, integrating the temporal inertial term over the control volume ω_f gives:

$$\int_{\omega_f} \frac{\partial \boldsymbol{\Phi}_{E,i}}{\partial t} = \sum_{g=1}^{N_f} \frac{\partial \boldsymbol{\Phi}_{E,i}^g}{\partial t} \int_{\omega_f} \phi_g = \mathbf{M}_h \frac{\partial \mathbf{U}_h}{\partial t}$$

where the mass matrix can be approximated by:

$$\forall g \leq N_{f0},\ (\mathbf{M}_h)_{f,g} = \int_{\omega_f} \phi_g \approx \text{meas}(\omega_f)\delta_{fg}$$

Thus the mass matrix $\mathbf{M}_h$ is diagonal and easy to reverse. As we are going to present in the next section, a diagonal mass matrix facilitates the computation of $\mathbf{B}_h \mathbf{M}_h^{-1} \mathbf{B}_h^t$ defined by (4.25) in the *prediction-correction* algorithm.

With $\mathbf{C}_h = \mathbf{B}_h^t$ and $\mathbf{E}_h = \mathbf{D}_h^t$, the Oseen local problem can be written in the matrix form:

$$\begin{cases} \mathbf{M}_h \dfrac{\partial \mathbf{U}_h}{\partial t} + (\mathbf{A}_h + \mathbf{L}_h(U_o))\,\mathbf{U}_h + \mathbf{B}_h^t \mathbf{P}_h + \mathbf{D}_h^t \boldsymbol{\Lambda}_h = \mathbf{0} \\ \mathbf{B}_h \mathbf{U}_h = \mathbf{G}_h \\ \mathbf{D}_h \mathbf{U}_h = \mathbf{H}_h \end{cases} \tag{4.21}$$

with $\mathbf{L}_h\left(\boldsymbol{U}_o\right)=\mathbf{0}$ for Stokes local problems (4.6)–(4.11).

4.3 The solution of local problems

The *prediction-correction* algorithm was originally proposed by [42, 135] for approximating in time the unsteady incompressible Navier–Stokes equations, then it was further developed in [76–80]. A major difficulty in the numerical simulation of incompressible flows is that the velocity and pressure are coupled by the incompressibility constraint. The *prediction-correction* algorithm is a time-marching technique composed of two sub-steps at each time step: (i) the prediction step where the pressure is treated explicitly (incremental *pressure-correction* scheme) or ignored (non-incremental *pressure-correction* scheme); (ii) the correction step where the velocity and pressure are corrected in order to verify the incompressibility constraint. Thus in this algorithm, at each time step, one needs to solve only a sequence of decoupled elliptic equations for the velocity and pressure.

However, the existence of Lagrange multipliers and the fact that $\mathbf{B}_h\mathbf{U}_h\neq\mathbf{0}$ makes (4.21) different from existing implementations in *TrioCFD*. Thus the original *prediction-correction* algorithm has been adapted to local problems in this thesis.

Now we present the solution of (4.21) by the incremental *prediction-correction* scheme with the backward Euler time discretization. We partition the time domain $(0,t_F)$ into N_t equally spaced intervals of length Δt and set $t^n=n\Delta t$. We denote the velocity and pressure at time step n by $\boldsymbol{U}_h^n$ and $\mathbf{P}_h^n$ and they are initialized by $\boldsymbol{U}_h^0=\mathbf{0}$ and $\mathbf{P}_h^0=\mathbf{0}$. The time step $n+1$ consists of the prediction step and the correction step which are described below.

Prediction step Using the velocity $\boldsymbol{U}_h^n$ and the pressure $\boldsymbol{P}_h^n$ of the precedent time step n, we solve the intermediate velocity $\boldsymbol{U}_h^*$ and the Lagrange multiplier $\boldsymbol{\Lambda}_h^*$ by

$$\begin{cases}\mathbf{M}_h\dfrac{\mathbf{U}_h^*-\mathbf{U}_h^n}{\Delta t}+\mathbf{A}_h\mathbf{U}_h^*+\mathbf{L}_h\left(\boldsymbol{U}_o\right)\mathbf{U}_h^*+\mathbf{D}_h^t\boldsymbol{\Lambda}_h^*=-\mathbf{B}_h^t\mathbf{P}_h^n\\ \mathbf{D}_h\mathbf{U}_h^*=\mathbf{H}_h\end{cases}$$

which is equivalent to

$$\begin{pmatrix}\dfrac{\mathbf{M}_h}{\Delta t}+\mathbf{A}_h+\mathbf{L}_h\left(\boldsymbol{U}_o\right) & \mathbf{D}_h^t\\ \mathbf{D}_h & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{U}_h^*\\ \boldsymbol{\Lambda}_h^*\end{pmatrix}=\begin{pmatrix}\dfrac{\mathbf{M}_h}{\Delta t}\mathbf{U}_h^n-\mathbf{B}_h^t\mathbf{P}_h^n\\ \mathbf{H}_h\end{pmatrix}\tag{4.22}$$

This is a typical saddle-point problem for which the original solvers and preconditioners in *TrioCFD* are no longer suitable. Thus we have implemented the *Schur complement* method using some functions in PETSc [122]. For the sake of simplicity, we denote

$$\tilde{\mathbf{A}}_h=\frac{\mathbf{M}_h}{\Delta t}+\mathbf{A}_h+\mathbf{L}_h\left(\boldsymbol{U}_o\right),\quad \mathbf{R}_h=\frac{\mathbf{M}_h}{\Delta t}\mathbf{U}_h^n-\mathbf{B}_h^t\mathbf{P}_h^n$$

Theoretically *Schur complement* solves (4.22) in three steps:

$$\begin{aligned}\tilde{\mathbf{A}}_h\mathbf{X}_h&=\mathbf{R}_h &&\Rightarrow \mathbf{X}_h\\ \mathbf{D}_h\left(\tilde{\mathbf{A}}_h\right)^{-1}\mathbf{D}_h^t\boldsymbol{\Lambda}_h^*&=\mathbf{D}_h\mathbf{X}_h-\mathbf{H}_h &&\Rightarrow \boldsymbol{\Lambda}_h^*\\ \tilde{\mathbf{A}}_h\mathbf{U}_h^*&=\mathbf{R}_h-\mathbf{D}_h^t\boldsymbol{\Lambda}_h^* &&\Rightarrow \mathbf{U}_h^*\end{aligned}$$

In practice, the Schur complement is not actually formed in PETSc [122]. Rather, only the matrix-vector product is performed by using the formula $\mathbf{S}_h=\mathbf{D}_h\left(\tilde{\mathbf{A}}_h\right)^{-1}\mathbf{D}_h^t$ and a solver is used to approximate the action of $\left(\tilde{\mathbf{A}}_h\right)^{-1}$. For more details, the reader can refer to the documentation of PETSc.

Correction step Denoting $\mathbf{P}'_h = \mathbf{P}_h^{n+1} - \mathbf{P}_h^n$ and using $\mathbf{U}_h^*$ solved in the last step, we solve $\left(\mathbf{U}_h^{n+1}, \mathbf{P}'_h\right)$ by

$$\mathbf{M}_h \frac{\boldsymbol{U}_h^{n+1} - \boldsymbol{U}_h^*}{\Delta t} + \mathbf{B}_h^t \mathbf{P}'_h = \mathbf{0} \tag{4.23}$$

$$\mathbf{B}_h \boldsymbol{U}_h^{n+1} = \mathbf{G}_h \tag{4.24}$$

Multiplying (4.23) by $\mathbf{B}_h$ and subtracting (4.24) from it, we obtain

$$\mathbf{B}_h \mathbf{M}_h^{-1} \mathbf{B}_h^t \mathbf{P}'_h = \frac{1}{\Delta t} \left(\mathbf{B}_h \mathbf{U}_h^* - \mathbf{G}_h\right) \tag{4.25}$$

Finally, the velocity and pressure at the end of the time step $n+1$ are computed using $\mathbf{U}_h^*$ and $\mathbf{P}'_h$ by

$$\begin{cases} \mathbf{U}_h^{n+1} = \mathbf{U}_h^* - \Delta t \mathbf{M}_h^{-1} \mathbf{B}_h^t \mathbf{P}'_h \\ \mathbf{P}_h^{n+1} = \mathbf{P}_h^n + \mathbf{P}'_h \end{cases}$$

4.4 The discretization of coarse-scale problems

We denote by V_H and M_H the multiscale approximation spaces defined in Chapter 3, for example, $V_H = V_H^{St}$ defined by (3.12) or $V_H = V_H^{Os}$ defined by (3.77) and M_H is defined by (3.11). The case of V_H and M_H defined by high-order multiscale methods will be presented specially in section 4.6. We recall the coarse-scale Navier-Stokes problem: find $(\boldsymbol{u}_H, \bar{p}_H) \in V_H \times M_H$ such that

$$\begin{aligned} -\mu \Delta \boldsymbol{u}_H + \rho \left(\boldsymbol{u}_H \cdot \nabla\right) \boldsymbol{u}_H + \nabla \bar{p}_H &= \boldsymbol{f} \text{ in } \Omega^\varepsilon \\ \text{div } \boldsymbol{u}_H &= 0 \text{ in } \Omega^\varepsilon \\ \boldsymbol{u}_H &= \boldsymbol{u}_D \text{ on } \Gamma_D \\ \mu \nabla \boldsymbol{u}_H \boldsymbol{n} - p_H \boldsymbol{n} &= \boldsymbol{h} \text{ on } \Gamma_N \end{aligned}$$

where $\partial\Omega = \Gamma_D \cup \Gamma_N$ with Γ_D and Γ_N the Dirichlet and the Neumann boundary respectively. $\boldsymbol{f}$ is a given volume force and $\boldsymbol{h}$ is a given surface force. Throughout this section, we assume that both μ and ρ are constant on the entire domain Ω^ε.

The nonlinear convective term makes the discretization of the coarse-scale Navier-Stokes problem complicated. Therefore we first choose the coarse-scale Oseen problem as a demonstrative example to show the discretization process. Besides, Oseen problems become Stokes problems when $\boldsymbol{U}_o = \mathbf{0}$. The discretization of the nonlinear convection term in coarse-scale Navier-Stokes problems will be presented specially in subsection 4.4.3.

The discrete variational formulation of the coarse-scale Oseen problem is: find $(\boldsymbol{u}_H, \bar{p}_H) \in V_H \times M_H$ such that

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \left(\mu\nabla \boldsymbol{u}_H : \nabla \boldsymbol{v} + \rho\left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{u}_H\right)\boldsymbol{v}\right) - \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \bar{p}_H \text{div } \boldsymbol{v} = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \boldsymbol{f}\cdot\boldsymbol{v} + \int_{\Gamma_N} \boldsymbol{h}\cdot\boldsymbol{v},\ \forall \boldsymbol{v} \in V_H \tag{4.26}$$

$$\sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} q \text{ div } \boldsymbol{u}_H = 0,\ \forall q \in M_H \tag{4.27}$$

We assume that the Oseen velocity $\boldsymbol{U}_o$ is constant in the entire domain Ω^ε and thus div $\boldsymbol{U}_o = 0$. We present only the discretization of coarse-scale problems in two dimensions in what follows.

4.4.1 Discretization of coarse-scale Oseen problems

Now we present the discretization of (4.26)–(4.27) using the Galerkin method specially developed in this thesis for multiscale finite element methods. In this Galerkin method, basis functions are the solutions of local problems. We denote the number of coarse elements in the coarse mesh $\mathcal{T}_H$ by N_T and the number of faces by N_F. As presented in Chapter 3, any function of V_H and M_H can be decomposed using multiscale basis functions as:

$$\forall \boldsymbol{u}_H \in V_H,\ \boldsymbol{u}_H = \sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s} u_{E,i}\boldsymbol{\Phi}_{E,i}, \quad \forall \bar{p}_H \in M_H,\ \bar{p}_H = \sum_{T\in\mathcal{T}_H} \bar{p}_T \Psi_T$$

where Ψ_T is the characteristic function of T ($\Psi_T = 1$ in T and 0 elsewhere) for $T \in \mathcal{T}_H$.

Substituting $\boldsymbol{u}_H$ and $\bar{p}_H$ into (4.26)–(4.27), taking $\boldsymbol{v} = \boldsymbol{\Phi}_{F,j}$ for $F \in \mathcal{E}_H$, and $j = 1, \ldots, s$ and $q = \Psi_T$ for $T \in \mathcal{T}_H$, we obtain

$$\sum_{T\in\mathcal{T}_H}\sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s} u_{E,i} \int_{T\cap\Omega^\varepsilon} \left(\mu\nabla\boldsymbol{\Phi}_{E,i} : \nabla\boldsymbol{\Phi}_{F,j} + \rho\left(\boldsymbol{U}_o\cdot\nabla\boldsymbol{\Phi}_{E,i}\right)\boldsymbol{\Phi}_{F,j}\right) - \sum_{T} \bar{p}_T \int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{\Phi}_{F,j} = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \boldsymbol{f}\cdot\boldsymbol{\Phi}_{F,j} + \int_{\Gamma_N} \boldsymbol{h}\cdot\boldsymbol{\Phi}_{F,j},$$

$$\sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s} u_{E,i} \int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{\Phi}_{E,i} = 0.$$

This problem can be written in the matrix form:

$$\begin{pmatrix} \mathbf{A}_H + \mathbf{L}_H(\boldsymbol{U}_o) & \mathbf{B}_H^t \\ \mathbf{B}_H & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}_H \\ \mathbf{P}_H \end{pmatrix} = \begin{pmatrix} \mathbf{F}_H \\ \mathbf{0} \end{pmatrix} \tag{4.28}$$

where

- $\mathbf{U}_H$ and $\mathbf{P}_H$ are vectors of size $N_F \times s$ and N_T respectively.

$$\mathbf{U}_H = \left(u_{E,1}, \cdots, u_{E,s}\right)_{E\in\mathcal{E}_H}, \quad \mathbf{P}_H = \left(\bar{p}_T\right)_{T\in\mathcal{T}_H}$$

- $\mathbf{A}_H$ and $\mathbf{L}_H(\boldsymbol{U}_o)$ are matrices with $N_F \times s$ rows and $N_F \times s$ columns.

$$(\mathbf{A}_H)_{Ei,Fj} = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \mu \nabla \boldsymbol{\Phi}_{E,i} : \nabla \boldsymbol{\Phi}_{F,j}, \text{ for } E, F \in \mathcal{E}_H,\ i,j = 1,\cdots,s$$

$$(\mathbf{L}_H(\boldsymbol{U}_o))_{Ei,Fj} = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \rho \left(\boldsymbol{U}_o \cdot \nabla \boldsymbol{\Phi}_{E,i}\right) \boldsymbol{\Phi}_{F,j}, \text{ for } E, F \in \mathcal{E}_H,\ i,j = 1,\cdots,s$$

- $\mathbf{B}_H$ is a matrix with N_T rows and $N_F \times s$ columns.

$$(\mathbf{B}_H)_{T,Ei} = \int_{T\cap\Omega^\varepsilon} \text{div } \boldsymbol{\Phi}_{E,i}, \text{ for } T \in \mathcal{T}_H,\ E \in \mathcal{E}_H,\ i = 1,\cdots,s$$

- $\mathbf{F}_H$ is a vector of size $N_F \times s$.

$$(\mathbf{F}_H)_{Fj} = \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \boldsymbol{f} \cdot \boldsymbol{\Phi}_{F,j} + \int_{\Gamma_N} \boldsymbol{h} \cdot \boldsymbol{\Phi}_{F,j}, \text{ for } F \in \mathcal{E}_H,\ j = 1,\cdots,s$$

4.4.2 The computation of matrices

In multiscale finite element methods, matrices $\mathbf{A}_H$, $\mathbf{B}_H$ and $\mathbf{L}_H(\boldsymbol{U}_o)$ are computed locally on coarse elements and then assembled on the coarse mesh. Given a coarse element $T \in \mathcal{T}_H$, given any multiscale basis function $\boldsymbol{\Phi}_{E,i}$ for $E \in \mathcal{E}(T)$ and $i = 1,\cdots,s$, by construction, $\boldsymbol{\Phi}_{E,i} \in V_h(T)$ defined by (4.4). Thus $\boldsymbol{\Phi}_{E,i}$ can be written in the Crouzeix-Raviart finite element basis functions as:

$$\boldsymbol{\Phi}_{E,i} = \sum_{m=1}^{N_f} \boldsymbol{\Phi}_{E,i}^m \phi_m,\ \forall E \in \mathcal{E}(T),\ i = 1,\cdots,s$$

In what follows, we make use of the properties of the Crouzeix-Raviart finite basis functions to compute integrals in the coefficients of matrices $\mathbf{A}_H$, $\mathbf{B}_H$ and $\mathbf{L}_H(\boldsymbol{U}_o)$.

Matrix $\mathbf{A}_H$ For $E, F \in \mathcal{E}_H$ and $i,j = 1,\cdots,s$:

$$\begin{aligned}(\mathbf{A}_H)_{Ei,Fj} &= \sum_{T\in\mathcal{T}_H} \int_{T\cap\Omega^\varepsilon} \mu \nabla \boldsymbol{\Phi}_{E,i} : \nabla \boldsymbol{\Phi}_{F,j} \\ &= \sum_{T\in\mathcal{T}_H} \sum_{m,n=1}^{N_f} \boldsymbol{\Phi}_{E,i}^m \cdot \boldsymbol{\Phi}_{F,j}^n \int_{T\cap\Omega^\varepsilon} \mu \nabla \phi_m \cdot \nabla \phi_n \\ &= \sum_{T\in\mathcal{T}_H} \sum_{m,n=1}^{N_f} \boldsymbol{\Phi}_{E,i}^m \cdot \boldsymbol{\Phi}_{F,j}^n \sum_{k=1}^{N_k} \int_{k\cap T\cap\Omega^\varepsilon} \mu \nabla \phi_m \cdot \nabla \phi_n\end{aligned}$$

Making use of equation (4.15), it is easy to deduce that

$$\int_{k\cap T\cap\Omega^\varepsilon} \mu \nabla \phi_m \cdot \nabla \phi_n = \mu \frac{\boldsymbol{S}_k^m \cdot \boldsymbol{S}_k^n}{\text{meas}(k)}$$

We recall that $\boldsymbol{S}_k^m$ and $\boldsymbol{S}_k^n$ are defined by (4.2).

Finally, the coefficient of $\mathbf{A}_H$ is defined by:

$$(\mathbf{A}_H)_{Ei,Fj} = \sum_{T\in\mathcal{T}_H} \sum_{m,n=1}^{N_f} \mathbf{\Phi}^m_{E,i}\cdot\mathbf{\Phi}^n_{F,j} \sum_{k=1}^{N_k} \mu \frac{\boldsymbol{S}^m_k\cdot\boldsymbol{S}^n_k}{\text{meas}(k)},\ \forall E,F\in\mathcal{E}_H,\ i,j=1,\cdots,s \tag{4.29}$$

Matrix $\mathbf{L}_H\left(\boldsymbol{U}_o\right)$ Denoting $\boldsymbol{U}_o = (\boldsymbol{U}^x_o, \boldsymbol{U}^y_o)$, for $E,F\in\mathcal{E}_H$ and $i,j=1,\cdots,s$:

$$\begin{aligned}(\mathbf{L}_H\left(\boldsymbol{U}_o\right))_{Ei,Fj} &= \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} \rho\left(\boldsymbol{U}_o\cdot\nabla\mathbf{\Phi}_{F,j}\right)\mathbf{\Phi}_{E,i}\\ &= \sum_{T\in\mathcal{T}_H}\sum_{m,n=1}^{N_f}\left(\mathbf{\Phi}^m_{F,j}\cdot\mathbf{\Phi}^n_{E,i}\right)\int_{T\cap\Omega^\varepsilon}\rho\left(\boldsymbol{U}^x_o\frac{\partial\phi_m}{\partial x}+\boldsymbol{U}^y_o\frac{\partial\phi_m}{\partial y}\right)\phi_n\end{aligned}$$

Making use of equation (4.15), we deduce that

$$\int_{T\cap\Omega^\varepsilon}\frac{\partial\phi_m}{\partial x}\phi_n = \sum_{k=1}^{N_k}\frac{\boldsymbol{S}^m_k\cdot\boldsymbol{e}_x}{\text{meas}(k)}\int_{k\cap T\cap\Omega^\varepsilon}\phi_n$$

In two dimensions, it is known that

$$\int_{k\cap T\cap\Omega^\varepsilon}\phi_n = \sum_{l=1}^{3}\frac{\text{meas}(k)}{3}\phi_n\left(\boldsymbol{s}_l\right)$$

where $\boldsymbol{s}_1,\cdots,\boldsymbol{s}_3$ are the three vertices of the element k.

Finally, the coefficient of $\mathbf{L}_H\left(\boldsymbol{U}_o\right)$ is defined by: $\forall E,F\in\mathcal{E}_H$ and $i,j=1,\cdots,s$:

$$(\mathbf{L}_H\left(\boldsymbol{U}_o\right))_{Ei,Fj} = \sum_{T\in\mathcal{T}_H}\sum_{k=1}^{N_k}\sum_{m,n\in k}\rho\left(\mathbf{\Phi}^m_{F,j}\cdot\mathbf{\Phi}^n_{E,i}\right)\frac{\left(\boldsymbol{U}_o\cdot\boldsymbol{S}^m_k\right)}{3}\sum_{l=1}^{3}\phi_n\left(\boldsymbol{s}_l\right)$$

Matrix $\mathbf{B}_H$ For $T\in\mathcal{T}_H$, for $E\in\mathcal{E}_H$ and $i=1,\cdots,s$:

$$(\mathbf{B}_H)_{T,Ei} = \int_{T\cap\Omega^\varepsilon}\text{div }\mathbf{\Phi}_{E,i} = \sum_{m=1}^{N_f}\mathbf{\Phi}^m_{E,i}\sum_{k=1}^{N_k}\int_{k\cap T\cap\Omega^\varepsilon}\text{div }\phi_m = \sum_{m=1}^{N_f}\mathbf{\Phi}^m_{E,i}\sum_{k=1}^{N_k}\boldsymbol{S}^m_k$$

For an internal face m shared by two adjacent elements k_1 and k_2, (4.2) reveals that $\boldsymbol{S}^m_{k_1}+\boldsymbol{S}^m_{k_2}=\mathbf{0}$. Consequently, the coefficient of $\mathbf{B}_H$ reduce to:

$$(\mathbf{B}_H)_{T,Ei} = \sum_{m\in\partial k\cap\Gamma_a}\mathbf{\Phi}^m_{E,i}\cdot\boldsymbol{S}^m_k,\ \forall T\in\mathcal{T}_H,\ E\in\mathcal{E}(T),\ i=1,\cdots,s \tag{4.30}$$

4.4.3 Discretization of the nonlinear convection term

In the initial stage of this thesis, we used the *prediction-correction* algorithm presented in section 4.3 to solve coarse-scale problems. This algorithm is a time-marching technique for unsteady or steady Navier-Stokes problems through a transient process. We partition the time domain $(0,t_F)$ into equally spaced intervals of length Δt and set for $n>0$, $t^n=n\Delta t$. At t^{n+1}, the nonlinear convective term of coarse-scale Naiver-Stokes problems is approximated by

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\rho\left(\boldsymbol{u}^{n+1}_H\cdot\nabla\right)\boldsymbol{u}^{n+1}_H\cdot\boldsymbol{v}\simeq\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\rho\left(\boldsymbol{u}^{n}_H\cdot\nabla\right)\boldsymbol{u}^{n+1}_H\cdot\boldsymbol{v},\ \forall\boldsymbol{v}\in V_H \tag{4.31}$$

Decomposing $\boldsymbol{u}_H^n$ and $\boldsymbol{u}_H^{n+1}$ using multiscale basis functions as:

$$\boldsymbol{u}_H^n = \sum_{G\in\mathcal{E}_H}\sum_{k=1}^{s} u_{G,k}^n \boldsymbol{\Phi}_{G,k}, \quad \boldsymbol{u}_H^{n+1} = \sum_{F\in\mathcal{E}_H}\sum_{j=1}^{s} u_{F,j}^{n+1} \boldsymbol{\Phi}_{F,j} \tag{4.32}$$

Substituting $\boldsymbol{u}_H^{n+1}$ defined by (4.32) into the convective term in the right hand side of (4.31) and taking $\boldsymbol{v} = \boldsymbol{\Phi}_{E,i}$ for $E \in \mathcal{E}_H$ and $i = 1, \cdots, s$, we obtain

$$\begin{aligned}\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon} \rho\left(\boldsymbol{u}_H^n\cdot\nabla\right)\boldsymbol{u}_H^{n+1}\cdot\boldsymbol{\Phi}_{E,i} &= \sum_{F\in\mathcal{E}_H}\sum_{j=1}^{s} u_{F,j}^{n+1}\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\rho\left(\boldsymbol{u}_H^n\cdot\nabla\right)\boldsymbol{\Phi}_{F,j}\cdot\boldsymbol{\Phi}_{E,i}\\ &= \sum_{F\in\mathcal{E}_H}\sum_{j=1}^{s}\mathbf{L}_{Ei,Fj}^n u_{F,j}^{n+1}\end{aligned} \tag{4.33}$$

where

$$\mathbf{L}_{Ei,Fj}^n = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\rho\left(\boldsymbol{u}_H^n\cdot\nabla\right)\boldsymbol{\Phi}_{F,j}\cdot\boldsymbol{\Phi}_{E,i},\ \forall F\in\mathcal{E}_H,\ j=1,\cdots,s$$

We notice that $\mathbf{L}_{Ei,Fj}^n$ contains $\boldsymbol{u}_H^n$ which is the velocity of the precedent time step n computed on the coarse mesh. Then substituting $\boldsymbol{u}_H^n$ defined by (4.32) into $\mathbf{L}_{Ei,Fj}^n$, we obtain

$$\mathbf{L}_{Ei,Fj}^n = \sum_{T\in\mathcal{T}_H}\sum_{G\in\mathcal{E}_H}\sum_{k=1}^{s} u_{G,k}^n\int_{T\cap\Omega^\varepsilon}\rho\left(\boldsymbol{\Phi}_{G,k}\cdot\nabla\right)\boldsymbol{\Phi}_{F,j}\cdot\boldsymbol{\Phi}_{E,i} = \sum_{T\in\mathcal{T}_H}\sum_{G\in\mathcal{E}_H}\sum_{k=1}^{s}\mathbf{L}_{Ei,Fj,Gk}^T u_{G,k}^n$$

where

$$\mathbf{L}_{Ei,Fj,Gk}^T = \int_{T\cap\Omega^\varepsilon}\rho\left(\boldsymbol{\Phi}_{G,k}\cdot\nabla\right)\boldsymbol{\Phi}_{F,j}\cdot\boldsymbol{\Phi}_{E,i},\ \forall E,F,G\in\mathcal{E}_H,\ i,j,k=1,\cdots,s. \tag{4.34}$$

We compute the term $\mathbf{L}_{Ei,Fj,Gk}^T$ locally on each coarse element $T \in \mathcal{T}_H$. Then when solving coarse-scale problems on the coarse mesh, at the time step $n+1$, we assemble matrix $\mathbf{L}_{Ei,Fj}^n$ using $\mathbf{L}_{Ei,Fj,Gk}^T$ and $u_{G,k}^n$ for $G \in \mathcal{E}_H$ and $k = 1, \cdots, s$.

Now we present how to compute $\mathbf{L}_{Ei,Fj,Gk}^T$ in two dimensions. We denote by $\{\boldsymbol{e}_x, \boldsymbol{e}_y\}$ the canonical basis of $\mathbb{R}^2$. On each coarse element $T \in \mathcal{T}_H$, by construction, for any $E \in \mathcal{E}(T)$ and $i = 1, \cdots, s$, multiscale basis function $\boldsymbol{\Phi}_{E,i} \in V_h(T)$. We can decompose multiscale basis functions in (4.34) with the Crouzeix-Raviart finite element basis functions: for any $E, F, G \in \mathcal{E}_H$ and $i, j, k = 1, \cdots, s$,

$$\begin{cases}\boldsymbol{\Phi}_{G,k} = \left(\boldsymbol{\Phi}_{G,k}^x, \boldsymbol{\Phi}_{G,k}^y\right) = \sum_{f_1=1}^{N_f}\left(\Phi_{G,k}^{x,f_1}, \Phi_{G,k}^{y,f_1}\right)\phi_{f_1}\\ \boldsymbol{\Phi}_{F,j} = \left(\boldsymbol{\Phi}_{F,j}^x, \boldsymbol{\Phi}_{F,j}^y\right) = \sum_{f_2=1}^{N_f}\left(\Phi_{F,j}^{x,f_2}, \Phi_{F,j}^{y,f_2}\right)\phi_{f_2}\\ \boldsymbol{\Phi}_{E,i} = \left(\boldsymbol{\Phi}_{E,i}^x, \boldsymbol{\Phi}_{E,i}^y\right) = \sum_{f_3=1}^{N_f}\left(\Phi_{E,i}^{x,f_3}, \Phi_{E,i}^{y,f_3}\right)\phi_{f_3}\end{cases} \tag{4.35}$$

In two dimensions, the matrix $\mathbf{L}^T_{Ei,Fj,Gk}$ defined by (4.34) equals to:

$$\begin{aligned}\mathbf{L}^T_{Ei,Fj,Gk} = &\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\rho\left(\Phi^x_{G,k}\frac{\partial\Phi^x_{F,j}}{\partial x}+\Phi^y_{G,k}\frac{\partial\Phi^x_{F,j}}{\partial y}\right)\Phi^x_{E,i}\\ &+\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\rho\left(\Phi^x_{G,k}\frac{\partial\Phi^y_{F,j}}{\partial x}+\Phi^y_{G,k}\frac{\partial\Phi^y_{F,j}}{\partial y}\right)\Phi^y_{E,i}\end{aligned}$$

Substituting (4.35) into this equality, we obtain

$$\begin{aligned}\mathbf{L}^T_{Ei,Fj,Gk} = &\sum_{f_1=1}^{N_f}\sum_{f_2=1}^{N_f}\sum_{f_3=1}^{N_f}\left(\Phi^{x,f_1}_{G,k}\Phi^{x,f_2}_{F,j}\Phi^{x,f_3}_{E,i}+\Phi^{x,f_1}_{G,k}\Phi^{y,f_2}_{F,j}\Phi^{y,f_3}_{E,i}\right)\int_{T\cap\Omega^\varepsilon}\rho\phi_{f_1}\frac{\partial\phi_{f_2}}{\partial x}\phi_{f_3}\\ &+\sum_{f_1=1}^{N_f}\sum_{f_2=1}^{N_f}\sum_{f_3=1}^{N_f}\left(\Phi^{y,f_1}_{G,k}\Phi^{x,f_2}_{F,j}\Phi^{x,f_3}_{E,i}+\Phi^{y,f_1}_{G,k}\Phi^{y,f_2}_{F,j}\Phi^{y,f_3}_{E,i}\right)\int_{T\cap\Omega^\varepsilon}\rho\phi_{f_1}\frac{\partial\phi_{f_2}}{\partial y}\phi_{f_3}\end{aligned}$$

For any element $k\in\mathcal{T}_h(T)$, for any $f_2\in\partial k$, $\nabla\phi_{f_2}$ is constant inside the element k and (4.15) implies that

$$\nabla\phi_{f_2}|_k = \frac{\boldsymbol{S}^{f_2}_k}{\operatorname{meas}(k)}$$

Since $\phi_{f_1}\phi_{f_3}$ is a quadratic polynomial, it is known in two dimensions that

$$\int_k\phi_{f_1}\phi_{f_3} = \begin{cases}\operatorname{meas}(k)/3, & if\ f_1=f_3 \text{ with } f_1,\ f_3\in\partial k\\ 0, & else\end{cases}$$

Thus it is easy to check that

$$\int_{T\cap\Omega^\varepsilon}\phi_{f_1}\frac{\partial\phi_{f_2}}{\partial x}\phi_{f_3} = \sum_{k\in T}\frac{\boldsymbol{S}^{f_2}_k\cdot\boldsymbol{e}_x}{\operatorname{meas}(k)}\int_{k\cap T\cap\Omega^\varepsilon}\phi_{f_1}\phi_{f_3} = \sum_{k\in T}\frac{\boldsymbol{S}^{f_2}_k\cdot\boldsymbol{e}_x}{3}$$

Finally, for $E,F,G\in\mathcal{E}(T)$ and $i,j,k=1,\cdots,s$, the matrix $\mathbf{L}^T_{Ei,Fj,Gk}$ is computed by:

$$\begin{aligned}\mathbf{L}^T_{Ei,Fj,Gk} = &\frac{1}{3}\sum_{k\in T}\sum_{f_1,f_2\in k}\rho\left(\Phi^{x,f_1}_{G,k}\Phi^{x,f_1}_{E,i}\Phi^{x,f_2}_{F,j}\boldsymbol{S}^{f_2}_k\cdot\boldsymbol{e}_x+\Phi^{x,f_1}_{G,k}\Phi^{y,f_1}_{E,i}\Phi^{y,f_2}_{F,j}\boldsymbol{S}^{f_2}_k\cdot\boldsymbol{e}_x\right)\\ &+\frac{1}{3}\sum_{k\in T}\sum_{f_1,f_2\in k}\rho\left(\Phi^{y,f_1}_{G,k}\Phi^{x,f_1}_{E,i}\Phi^{x,f_2}_{F,j}\boldsymbol{S}^{f_2}_k\cdot\boldsymbol{e}_y+\Phi^{y,f_1}_{G,k}\Phi^{y,f_1}_{E,i}\Phi^{y,f_2}_{F,j}\boldsymbol{S}^{f_2}_k\cdot\boldsymbol{e}_y\right)\end{aligned}$$

with $\boldsymbol{S}^{f_2}_k$ defined by (4.2).

The discretization of the SUPG stabilization term is very similar to that of the nonlinear convection term and will not be repeated here.

4.5 The solution of coarse-scale problems

We solved coarse-scale problems by the *prediction-correction* algorithm (see section 4.3) in the early stage of this thesis. We recall that this algorithm is a time-marching technique for unsteady Navier-Stokes equations. This algorithm can also solve steady problems but via a pseudo-transient process. As we are going to present in section 4.6, we have implemented later in this thesis a direct solver to solve steady local and coarse-scale problems without passing through the transient process.

In the *prediction-correction* algorithm, the unsteady $\boldsymbol{u}_H$ and $\bar{p}_H$ can be written as

$$\begin{aligned}\forall \boldsymbol{u}_H \in V_H,\ \boldsymbol{u}_H(t) &= \sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s} u_{E,i}(t)\boldsymbol{\Phi}_{E,i}\\ \forall \bar{p}_H \in M_H,\ \bar{p}_H(t) &= \sum_{T\in\mathcal{T}_H} \bar{p}_T(t)\Psi_T\end{aligned}$$

then the temporal inertial term is discretized by

$$\sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\frac{\partial \boldsymbol{u}_H}{\partial t}\cdot\boldsymbol{\Phi}_{F,j} = \sum_{T\in\mathcal{T}_H}\sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s}\frac{\partial u_{E,i}}{\partial t}\int_{T\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\Phi}_{F,j},\ \forall F\in\mathcal{E}_H,\ j=1,\cdots,s$$

We decompose multiscale basis functions using the Crouzeix-Raviart finite element basis functions defined in subsection 4.1.1: for any $E,F\in\mathcal{E}_H$ and $i,j=1,\cdots,s$,

$$\boldsymbol{\Phi}_{E,i} = \sum_{m=1}^{N_f}\boldsymbol{\Phi}_{E,i}^{m}\phi_m,\quad \boldsymbol{\Phi}_{F,j} = \sum_{n=1}^{N_f}\boldsymbol{\Phi}_{F,j}^{n}\phi_n \tag{4.36}$$

Substituting (4.36) into the temporal inertial term and we obtain its matrix form:

$$\mathbf{M}_H\frac{\partial \mathbf{U}_H}{\partial t}$$

where the mass matrix $\mathbf{M}_H$ is computed by: for any $E,F\in\mathcal{E}_H$ and $i,j=1,\cdots,s$,

$$(\mathbf{M}_H)_{Ei,Fj} = \sum_{T\in\mathcal{T}_H}\int_{T\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\Phi}_{F,j} = \sum_{T\in\mathcal{T}_H}\sum_{k=1}^{N_k}\sum_{m=1}^{N_f}\boldsymbol{\Phi}_{E,i}^{m}\cdot\boldsymbol{\Phi}_{F,j}^{m}\frac{\mathrm{meas}(k)}{3}$$

The matrix $\mathbf{M}_H$ is sparse but might be non-diagonal, which makes it difficult to compute the pressure matrix $\mathbf{B}_H\mathbf{M}_H^{-1}\mathbf{B}_H^t$ (see (4.25)). In practice, we construct a condensed matrix which is diagonal to facilitate the computation of the pressure matrix. For example, we summarize absolute values of all coefficients in each row of the original matrix and take the sum as the diagonal coefficient of the condensed matrix. Consequently, the condensed matrix is a diagonal matrix. Finally the unsteady coarse-scale Oseen problem can be written in the matrix form:

$$\begin{aligned}\mathbf{M}_H\frac{\partial \mathbf{U}_H}{\partial t} + \left(\mathbf{A}_H + \mathbf{L}_H(\boldsymbol{U}_o)\right)\mathbf{U}_H + \mathbf{B}_H^t\mathbf{P}_H &= \mathbf{F}_H\\ \mathbf{B}_H\mathbf{U}_H &= \mathbf{0}\end{aligned}$$

The outflow condition $\nabla\boldsymbol{u}_H\cdot\boldsymbol{n} - \bar{p}_H\cdot\boldsymbol{n} = \boldsymbol{h}$ is a natural condition as it is included in the variational formulation (4.26). We approximate the non-homogeneous Dirichlet boundary condition $\boldsymbol{u}_H = \boldsymbol{u}_D$ in a weak form:

$$\int_E \boldsymbol{u}_H\cdot\boldsymbol{\omega}_{E,i} = \int_E \boldsymbol{u}_D\cdot\boldsymbol{\omega}_{E,i},\ \text{for all } E\in\mathcal{E}_H \text{ on } \Gamma_D,\ i=1,\cdots,s.$$

4.6 Technical aspects of high-order multiscale methods

Previously, we presented the discretization of local problems using the Crouzeix-Raviart finite element. For the high-order multiscale finite element methods, we discretize also local problems using the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element. The idea is to improve the final accuracy of the pressure given by high-order multiscale methods. For local and coarse-scale problems in high-order multiscale methods, we present a direct solver which is an alternative to the *prediction-correction*.

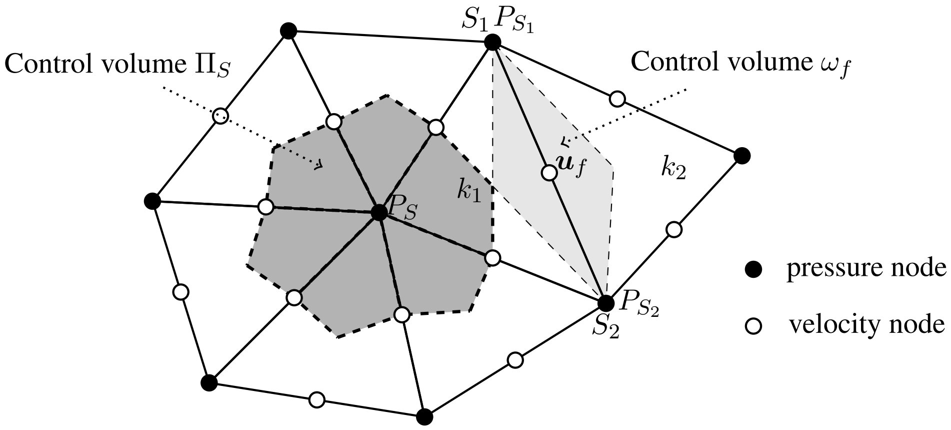


FIGURE 4.7: The $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element and control volumes

4.6.1 The $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element

In the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element [123], as shown in Figure 4.7, the velocity is located at the barycenter of faces and the pressure is located at the vertex of elements. The control volume of velocity ω_f is constructed the same as in the Crouzeix-Raviart element. The control volume of pressure Π_S is constructed by joining the barycenter of faces with the barycenter of all elements which share the pressure node P_S. For any $T \in \mathcal{T}_H$, we denote by N_S the number of pressure nodes in $\mathcal{T}_h(T)$. The finite element space of velocity $V_h(T)$ is the same as (4.4). We define the finite element space of pressure $M_h^s(T)$ by

$$M_h^S(T) = \left\{ q_h \in L^2\left(T \cap \Omega^\varepsilon\right) \cap C^0\left(T \cap \Omega^\varepsilon\right) \; \middle| \; q_h|_k \in \mathbb{P}_1\left(k\right), \; \forall k \in \mathcal{T}_h\left(T\right) \right\}$$

where $C^0\left(T \cap \Omega^\varepsilon\right)$ is the set of continuous functions on $T \cap \Omega^\varepsilon$.

Let $\{\psi_1, \cdots, \psi_{N_S}\}$ be a basis of $M_h^S(T)$ where ψ_m satisfies $\psi_m\left(\boldsymbol{x}_n\right) = \delta_{mn}$ for any $n = 1, \cdots, N_S$. We recall that $\left\{\phi_1, \cdots, \phi_{N_f}\right\}$ is a basis of each component of $V_h(T)$ where $\phi_i(x_j) = \delta_{ij}$ for $i, j = 1, \cdots, N_f$.

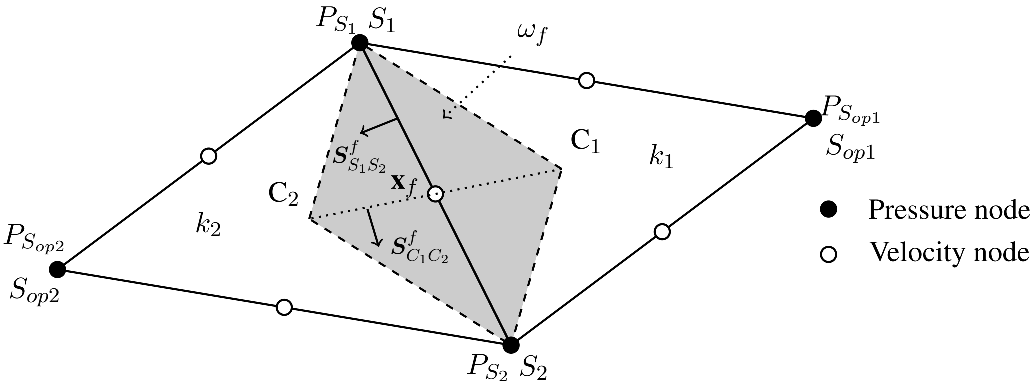


FIGURE 4.8: The control volume ω_f of an internal velocity node in the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element

4.6.2 The discretization of local problems

In two dimensions, we recall here local problems defined by (3.191). On any $T \in \mathcal{T}_H$, for each $E \in \mathcal{E}(T)$ and for $i = 1, \cdots, s$, find $\mathbf{\Phi}_{E,i} \in V_h(T)$, $\pi_{E,i} \in M_h^s(T)$ and $\boldsymbol{\eta}_{E,i} \in \mathbb{R}^2$ such that

$$-\mu \Delta \mathbf{\Phi}_{E,i} + \nabla \pi_{E,i} = \boldsymbol{\eta}_{E,i} \text{ in } T \cap \Omega^\varepsilon \tag{4.37}$$

$$\text{div } \mathbf{\Phi}_{E,i} \in \text{span}\left\{\varpi_{T,1}, \cdots, \varpi_{T,3}\right\} \text{ in } T \cap \Omega^\varepsilon \tag{4.38}$$

$$\mu \nabla \mathbf{\Phi}_{E,i} \boldsymbol{n} - \pi_{E,i} \boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\right\} \text{ on } F,\ \forall F \in \mathcal{E}(T) \tag{4.39}$$

$$\mathbf{\Phi}_{E,i} = \mathbf{0} \text{ on } \partial B^\varepsilon \cap T \tag{4.40}$$

$$\int_F \mathbf{\Phi}_{E,i} \cdot \boldsymbol{\omega}_{F,j} = \left\{ \begin{array}{ll} \delta_{ij}, & F = E \\ 0, & F \neq E \end{array} \right. \quad \forall F \in \mathcal{E}(T),\ j = 1, \cdots, s \tag{4.41}$$

$$\int_{T \cap \Omega^\varepsilon} \mathbf{\Phi}_{E,i} \cdot \boldsymbol{e}_l = 0\ \forall l = 1, \cdots, d \tag{4.42}$$

$$\int_{T \cap \Omega^\varepsilon} \pi_{E,i} \cdot \varpi_{T,m} = 0\ \forall m = 1, \cdots, t \tag{4.43}$$

where weighting functions are defined by (3.203). Throughout this section, we assume that μ is constant on $T \cap \Omega^\varepsilon$.

The discretization of this system using the Crouzeix-Raviart finite element is very similar to what was presented in section 4.2. Now we discretize this system using the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element. The pressure $\pi_{E,i}$ can be decomposed in the basis of $M_h^s(T)$ as

$$\forall E \in \mathcal{E}(T),\ i = 1, \cdots, s,\ \pi_{E,i} = \sum_{m=1}^{N_S} \pi_{E,i}^m \psi_m$$

and the discretized pressure is denoted by a vector $\mathbf{P}_h = \left(\pi_{E,i}^1, \cdots, \pi_{E,i}^{N_S}\right)$.

Definition 4.6.1. *As shown in Figure 4.8 and Figure 4.9, for an internal node $\boldsymbol{x}_f$, we use n_s as the index of vertex in the face f, i.e. S_1^f and S_2^f. We use n_{sop} as the index of vertex opposite to the face f, i.e. S_{op1} and S_{op2}. The vector $\boldsymbol{S}_{C_1C_2}$ and $\boldsymbol{S}_{S_1S_2}$ represent respectively the vector normal to segments $[C_1C_2]$ and $[S_1S_2]$ with measure* $\text{meas}([C_1C_2])$ *and* $\text{meas}([S_1S_2])$.

Conservation of momentum equation Since $\boldsymbol{\eta}_{E,i} = \left(\eta_{E,i}^1, \cdots, \eta_{E,i}^d\right) \in \mathbb{R}^d$, thus we denote the vector $\boldsymbol{\eta}_h = \left(\eta_{E,i}^1, \cdots, \eta_{E,i}^d\right)$. Equation (4.37) is discretized in the similar way as what was presented in subsection 4.2.2. Then (4.37) can be written in the matrix form:

$$\mathbf{A}_h \mathbf{U}_h + \tilde{\mathbf{C}}_h \mathbf{P}_h + \mathbf{E}_h \mathbf{\Lambda}_h + \mathbf{X}_h \boldsymbol{\eta}_h = \mathbf{F}_h \tag{4.44}$$

where $\mathbf{A}_h$ and $\mathbf{E}_h$ have the same formulas as what were defined in subsection 4.2.2 and we recall:

$$\mathbf{A}_h = \begin{pmatrix} A_h & 0 \\ 0 & A_h \end{pmatrix}$$

where

$$\forall f,\ g \leq N_{f0},\ (A_h)_{f,g} = -\int_{\gamma_f} \mu \nabla \phi_g \cdot \boldsymbol{n}$$

$$\forall f \leq N_{f0},\ F \in \mathcal{E}(T),\ j = 1, \cdots, s,\ (\mathbf{E}_h)_{f,(F,j)} = \int_{\partial \omega_f \cap F} \boldsymbol{\omega}_{F,j} \tag{4.45}$$

Since $\mathcal{E}(T)$ contains three boundaries, $\mathbf{E}_h$ is a matrix with $N_f \times d$ rows and $3 \times s$ columns.

It is easy to check that matrices $\tilde{\mathbf{C}}_h$ and $\mathbf{X}_h$ in (4.44) are defined by

$$\forall f \leq N_{f0},\ \ m \leq N_S,\ \left(\tilde{\mathbf{C}}_h\right)_{f,m} = \int_{\gamma_f} \psi_m \boldsymbol{n} \tag{4.46}$$

$$\forall f \leq N_{f0},\ l = 1, \cdots, d,\ \left(\mathbf{X}_h\right)_{f,l} = \text{meas}\left(\omega_f\right) \tag{4.47}$$

It is important to note that $\tilde{\mathbf{C}}_h$ is different from $\mathbf{C}_h$ defined by subsection 4.2.2 since the discretization of pressure is different. The computation of $\tilde{\mathbf{C}}_h$ is presented in detail in [123] and we give directly the coefficient of $\tilde{\mathbf{C}}_h$ here. Making use of Definition 4.6.1, the coefficient of $\tilde{\mathbf{C}}_h$ is

$$\forall f \leq N_{f0},\ \left(\tilde{\mathbf{C}}_h\right)_{f,n_s} = \frac{1}{2}\boldsymbol{S}^f_{C_1C_2} \tag{4.48}$$

$$\forall f \leq N_{f0},\ \left(\tilde{\mathbf{C}}_h\right)_{f,n_{sop}} = \frac{1}{6}\boldsymbol{S}^f_{S_1S_2} \tag{4.49}$$

Conservation of mass Equation (3.203) implies that weighting functions $\varpi_{T,m}$ for $m = 1, \cdots, t$ are

$$\varpi_{T,1} = 1,\ \varpi_{T,2} = x,\ \varpi_{T,3} = y$$

Equation (4.38) implies that there exist $\kappa_0, \kappa_1, \kappa_2 \in \mathbb{R}$ such that

$$\text{div } \boldsymbol{\Phi}_{E,i} = \kappa_0 + \kappa_1 x + \kappa_2 y \text{ in } T \cap \Omega^{\varepsilon}$$

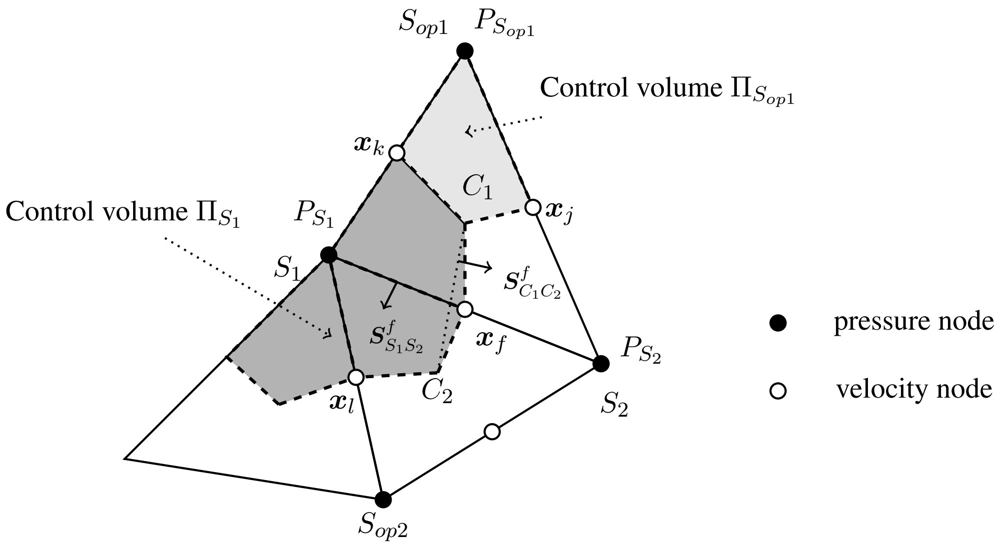


FIGURE 4.9: The control volumes of pressure nodes in the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element

Integrating this equation over the control volume Π_{S_1} shown in Figure 4.9 of the pressure node S_1 yields

$$\int_{\Pi_{S_1}} \text{div } \boldsymbol{\Phi}_{E,i} - \int_{\Pi_{S_1}} \left(\kappa_0 + \kappa_1 x + \kappa_2 y\right) = 0$$

We recall that $\mathbf{\Phi}_{E,i} = \mathbf{0}$ on Γ_d by formula (4.40). The equation above is equivalent to

$$\int_{\partial \Pi_{S_1} \backslash \Gamma_d} \mathbf{\Phi}_{E,i} \cdot \boldsymbol{n} = \int_{\Pi_{S_1}} (\kappa_0 + \kappa_1 x + \kappa_2 y)$$

Using similar arguments as those in subsection 4.2.1 and denoting the vector $\mathbf{K}_h = (\kappa_0, \kappa_1, \kappa_2)$, the equation above can be written in the matrix form:

$$\tilde{\mathbf{B}}_h \mathbf{U}_h + \mathbf{Y}_h \mathbf{K}_h = \mathbf{0} \tag{4.50}$$

where

$$\begin{aligned} &\forall S_1 \le N_S,\ f \le N_{f0},\ \left(\tilde{\mathbf{B}}_h\right)_{S_1,f} = -\int_{\partial \Pi_{S_1} \backslash \Gamma_d} \phi_f \boldsymbol{n} \\ &\forall S_1 \le N_S,\ (\mathbf{Y}_h)_{S_1,0} = \int_{\Pi_{S_1}} 1, \quad (\mathbf{Y}_h)_{S_1,1} = \int_{\Pi_{S_1}} x, \quad (\mathbf{Y}_h)_{S_1,2} = \int_{\Pi_{S_1}} y \end{aligned} \tag{4.51}$$

Note that we abuse the index by using S_1 also as the index of the pressure node S_1.

We compute the coefficient $(\mathbf{Y}_h)_{S_1,0}$ by

$$(\mathbf{Y}_h)_{S_1,0} = \int_{\Pi_{S_1}} 1 = \sum_{k \in \Pi_{S_1}} \frac{1}{3} \text{meas}\,(k)$$

The coefficients $(\mathbf{Y}_h)_{S_1,1}$ and $(\mathbf{Y}_h)_{S_1,2}$ can also be computed in the similar way.

Now let us compute the coefficient of $\tilde{\mathbf{B}}_h$. In the first step, we compute the contribution of ϕ_f on the control volume Π_{S_1}. Using appropriate orientations of vectors $\boldsymbol{S}_{fC_1}$, $\boldsymbol{S}_{fC_2}$, $\boldsymbol{S}_{kC_1}$ and $\boldsymbol{S}_{lC_1}$, we have

$$\left(\tilde{\mathbf{B}}_h\right)_{S_1,f} = -\int_{\partial \Pi_{S_1} \backslash \Gamma_d} \phi_f \boldsymbol{n} = \frac{1}{3} \left(\boldsymbol{S}_{fC_1} + \boldsymbol{S}_{fC_2}\right) + \frac{1}{3} \left(\boldsymbol{S}_{kC_1} + \boldsymbol{S}_{lC_1}\right) \tag{4.52}$$

Then Figure 4.9 shows that

$$\boldsymbol{S}_{kC_1} = \frac{1}{3} \boldsymbol{S}_{kS_2}, \quad \boldsymbol{S}_{lC_2} = \frac{1}{3} \boldsymbol{S}_{lS_2}$$

Besides, it is trivial to verify that

$$\boldsymbol{S}_{kS_2} + \boldsymbol{S}_{lS_2} = \boldsymbol{S}_{kl} = \frac{3}{2} \boldsymbol{S}^f_{C_1C_2}$$

Thus the coefficient of $\tilde{\mathbf{B}}_h$ corresponding to the contribution of ϕ_f in the control volume Π_{S_1} is:

$$\left(\tilde{\mathbf{B}}_h\right)_{S_1,f} = \frac{1}{2} \boldsymbol{S}^f_{C_1C_2}$$

Finally, making use of Definition 4.6.1, the coefficient of $\tilde{\mathbf{B}}_h$ corresponding to contribution of ϕ_f in the control volume of a vertex which is located in the face f is:

$$\forall f \le N_{f0},\ \left(\tilde{\mathbf{B}}_h\right)_{n_s,f} = \frac{1}{2} \boldsymbol{S}^f_{C_1C_2} \tag{4.53}$$

In the second step, let us compute the contribution of ϕ_f in the control volume $\Pi_{S_{op1}}$. As shown in Figure 4.9,

$$\left(\tilde{\mathbf{B}}_h\right)_{S_{op1},f} = \int_{\partial\Pi_{S_{op1}}\setminus\Gamma_d} \phi_f \boldsymbol{n} = \frac{1}{3}\left(\boldsymbol{S}_{jC_1} + \boldsymbol{S}_{C_1k}\right) = \frac{1}{3}\boldsymbol{S}_{jk}$$

It is easy to see that $\boldsymbol{S}_{jk} = \frac{1}{2}\boldsymbol{S}^f_{S_1S_2}$. Consequently, we have

$$\left(\tilde{\mathbf{B}}_h\right)_{S_{op1},f} = \frac{1}{6}\boldsymbol{S}^f_{S_1S_2}$$

Finally, making use of Definition 4.6.1, the coefficient of $\tilde{\mathbf{B}}_h$ corresponding to contribution of ϕ_f in the control volume of a vertex which is opposite to the face f is:

$$\forall f \leq N_{f0},\ \left(\tilde{\mathbf{B}}_h\right)_{n_{sop},f} = \frac{1}{6}\boldsymbol{S}^f_{S_1S_2} \tag{4.54}$$

Comparing coefficients of $\tilde{\mathbf{B}}_h$ defined by (4.53)–(4.54) with coefficients of $\tilde{\mathbf{C}}_h$ defined by (4.48)–(4.49), we conclude that $\tilde{\mathbf{C}}_h = \tilde{\mathbf{B}}^t_h$.

Discretization of the velocity integral boundary condition Equation (4.41) is discretized in the same way as what was presented in subsection 4.2.4 and

$$\mathbf{D}_h\mathbf{U}_h = \mathbf{H}_h \tag{4.55}$$

where

$$\forall F \in \mathcal{E}(T),\ j = 1,\cdots,s,\ f \leq N_{f0},\ (\mathbf{D}_h)_{(F,j),f} = \int_{F\cap\partial\omega_f} \boldsymbol{\omega}_{F,j}$$

$$\forall F \in \mathcal{E}(T),\ j = 1,\cdots,s,\ (\mathbf{H}_h)_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases}$$

Comparing to the coefficients of $\mathbf{E}_h$ defined by (4.45), it is easy to see that $\mathbf{E}_h = \mathbf{D}^t_h$.

Discretization of the integral of velcity and pressure It is easy to check that (4.42) and (4.43) can be written in the matrix form:

$$\mathbf{S}_h\mathbf{U}_h = \mathbf{0},\quad \mathbf{Q}_h\mathbf{P}_h = \mathbf{0} \tag{4.56}$$

where

$$\forall l = 1,\cdots,d,\ f \leq N_{f0},\ (\mathbf{S}_h)_{l,f} = \text{meas}\left(\omega_f\right)$$

$$\forall S_1 \leq N_S,\ (\mathbf{Q}_h)_{0,S_1} = \int_{T\cap\Omega^\varepsilon} \psi_{S_1},\quad (\mathbf{Q}_h)_{1,S_1} = \int_{T\cap\Omega^\varepsilon} x\psi_{S_1},\quad (\mathbf{Q}_h)_{2,S_1} = \int_{T\cap\Omega^\varepsilon} y\psi_{S_1}$$

Comparing to the coefficients of $\mathbf{X}_h$ defined by (4.47), it is trivial to verify that $\mathbf{X}_h = \mathbf{S}^t_h$. Now let us compute the coefficients of matrix $\mathbf{Q}_h$. For the pressure node S_1 and the control volume Π_{S_1} shown in Figure 4.9, the coefficient $(\mathbf{Q}_h)_{0,S_1}$ is computed by

$$(\mathbf{Q}_h)_{0,S_1} = \int_{T\cap\Omega^\varepsilon} \psi_{S_1} = \sum_{k\in\Pi_{S_1}} \int_k \psi_{S_1} = \sum_{k\in\Pi_{S_1}} \frac{1}{3}\text{meas}\,(k)$$

The coefficients $(\mathbf{Q}_h)_{1,S_1}$ and $(\mathbf{Q}_h)_{2,S_1}$ are approximated by

$$(\mathbf{Q}_h)_{1,S_1} = \int_{T\cap\Omega^\varepsilon} x\psi_{S_1} \approx \int_{\Pi_{S_1}} x$$

$$(\mathbf{Q}_h)_{2,S_1} = \int_{T\cap\Omega^\varepsilon} y\psi_{S_1} \approx \int_{\Pi_{S_1}} y$$

Comparing with the coefficients of $\mathbf{Y}_h$ defined by (4.51), we conclude that $\mathbf{Y}_h = \mathbf{Q}_h^t$.

Finally, local problem (4.37)–(4.43) can be written in the following matrix form:

$$\begin{pmatrix} \mathbf{A}_h & \mathbf{B}_h^t & \mathbf{D}_h^t & \mathbf{0} & \mathbf{S}_h^t \\ \mathbf{B}_h & \mathbf{0} & \mathbf{0} & \mathbf{Q}_h^t & \mathbf{0} \\ \mathbf{D}_h & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_h & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_h & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}_h \\ \mathbf{P}_h \\ \mathbf{\Lambda}_h \\ \mathbf{K}_h \\ \boldsymbol{\eta}_h \end{pmatrix} = \begin{pmatrix} \mathbf{F}_h \\ \mathbf{0} \\ \mathbf{H}_h \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \tag{4.57}$$

Remark. *It would be more accurate to solve (4.37)–(4.43) with the $\mathbb{P}_2/\mathbb{P}_1$ finite element. However, this finite element is not available in TrioCFD. Nevertheless, we were able to solve system (4.57) in Freefem++ [81] using the $\mathbb{P}_2/\mathbb{P}_1$ finite element. We prove the existence and uniqueness of a solution to system (4.57) and present the multiscale basis functions in Appendix B.*

4.6.3 The discretization of coarse-scale problems

We recall the coarse-scale Stokes problem (3.204)–(3.205) defined in high-order multiscale methods:

$$\sum_{T\in\Omega^\varepsilon}\int_{T\cap\Omega^\varepsilon} \mu\nabla\boldsymbol{u}_H : \nabla\boldsymbol{v} - \sum_{T\in\Omega^\varepsilon}\int_{T\cap\Omega^\varepsilon} \bar{p}_H \operatorname{div}\boldsymbol{v} = \sum_{T\in\Omega^\varepsilon}\int_{T\cap\Omega^\varepsilon} \boldsymbol{f}\cdot\boldsymbol{v},\ \boldsymbol{v}\in\widehat{V}_H^{St} \tag{4.58}$$

$$\sum_{T\in\Omega^\varepsilon}\int_{T\cap\Omega^\varepsilon} q \operatorname{div}\boldsymbol{u}_H = 0,\ \forall q\in\widehat{M}_H \tag{4.59}$$

The coarse-scale Oseen problems are discretized similarly to coarse-scale Stokes problems and the discretization is thus not detailed here. System (4.58)–(4.59) is discretized by the Galerkin method where basis functions are solutions of local problems defined in high-order Crouzeix-Raviart multiscale methods. In two dimensions, $d = 2$, we decompose the velocity $\boldsymbol{u}_H \in \widehat{V}_H^{St}$ defined by (3.167) and the pressure $\bar{p}_H \in \widehat{M}_H$ defined by (3.166) using multiscale basis functions as:

$$\boldsymbol{u}_H = \sum_{E\in\mathcal{E}_H}\sum_{i=1}^{s} u_{E,i}\boldsymbol{\Phi}_{E,i} + \sum_{T\in\mathcal{T}_H}\sum_{k=1}^{d} u_{T,k}\boldsymbol{\Psi}_{T,k}$$

$$\bar{p}_H = \sum_{T\in\mathcal{T}_H}\left(\bar{p}_0^T + \bar{p}_1^T x + \bar{p}_1^T y\right)\Theta_T$$

where Θ_T is the characteristic function of element T, i.e. $\Theta_T = 1$ in T and 0 elsewhere.

We denote the discretized velocity and pressure by

$$\mathbf{U}_H = (u_{E,1},\cdots,u_{E,s},\ u_{T,1},\cdots,u_{T,d})\ \text{for all } E\in\mathcal{E}_H,\ T\in\mathcal{T}_H$$

$$\mathbf{P}_H = \left(\bar{p}_0^T,\ \bar{p}_1^T,\ \bar{p}_2^T\right)_{T\in\mathcal{T}_H}$$

Note that vectors $\mathbf{U}_H$ and $\mathbf{P}_H$ are of sizes $(N_F\times s + N_T\times d)$ and $N_T\times 3$ respectively.

Substituting $\boldsymbol{u}_H$ and $\bar{p}_H$ into (4.58)–(4.59), choosing $\boldsymbol{v} = \boldsymbol{\Phi}_{F,j}$ for $F \in \mathcal{E}_H$ and $j = 1, \cdots, s$ or $\boldsymbol{v} = \boldsymbol{\Psi}_{T,k}$ for $k = 1, \cdots, d$, choosing $q = \Theta_T$ and proceeding similarly to what was presented in subsection 4.4.1, system (4.58)–(4.59) can be written in the matrix form:

$$\begin{pmatrix} \mathbf{A}_H & \mathbf{B}^t \\ \mathbf{B}_H & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}_H \\ \mathbf{P}_H \end{pmatrix} = \begin{pmatrix} \mathbf{F}_H \\ \mathbf{0} \end{pmatrix} \tag{4.60}$$

where $\mathbf{A}_H$, $\mathbf{B}_H$ and $\mathbf{F}_H$ are computed similarly to what is presented in section 4.4.

After obtaining $\mathbf{P}_H$, the fine-scale pressure is reconstructed on each coarse element by

$$\forall T \in \mathcal{T}_H, \; p_H|_T = \sum_{E \in \mathcal{E}(T)} \sum_{i=1}^{s} u_{E,i} \pi_{E,i} + \sum_{k=1}^{d} u_{T,k} \pi_{T,k} + \left(\bar{p}_0^T + \bar{p}_1^T x + \bar{p}_2^T y\right)$$

4.6.4 The solution of local and coarse-scale problems with a direct solver

In practice, we have remarked several limitations of the *prediction-correction* algorithm when solving the local problem (4.57) and the coarse-scale problem (4.60). Firstly, this algorithm solves steady flow problems through a pseudo-transient process which can be time-consuming. Secondly, this algorithm is designed to solve the pressure and the velocity separately in two steps in each time step. However, besides the pressure and the velocity, there exist some Lagrange multipliers in (4.57), thus it is difficult to adapt this algorithm to solve all these unknowns separately. Thirdly, in this algorithm, pressure does not have zero average over the computational domain and its uniqueness is fixed by the pressure imposed on the boundary of the domain. This is not compatible with (4.43), where the weighted average of pressure must be zero on the entire domain.

Since the local problem (4.57) and the coarse-scale problem (4.60) are steady problems and they are not of too large sizes, we used a direct solver *UMFPACK* [53] as an alternative to the *prediction-correction* algorithm. *UMFPACK* is a set of routines for solving unsymmetric sparse linear systems written in the form $Ax = b$, using the Unsymmetric-pattern MultiFrontal method and direct sparse LU factorization. Using this direct solver, physical unknowns are solved in a coupled way without passing through a pseudo-transient process. As a result, this solver is much less time-consuming than the *prediction-correction* algorithm when solving both local and coarse-scale problems. It should be noted that this direct solver was developed only for solving local problems and steady coarse-scale Stokes or Oseen problems in two dimensions in this thesis.

In the original implementation of *TrioCFD*, matrices $\mathbf{B}_h$ and $\mathbf{B}_h^t$ are not assembled, because only matrix-vector products $\mathbf{B}_h\mathbf{U}_h$ and $\mathbf{B}_h^t\mathbf{P}_h$ are required when solving (4.22) and (4.23)–(4.24) by iterative methods. In other words, matrix systems such as (4.57) or (4.60) are not assembled in the *prediction-correction* algorithm. Besides, the architect of the software is designed only for the *prediction-correction* algorithm. Therefore an important work was devoted to the assembly of matrix systems (4.57) and (4.60) as well as to the adaptation of the architecture to the direct solver.

4.6.5 The validation of solutions of local problems

In order to validate the implementation in *TrioCFD* of local problems (4.37)–(4.43) defined in high-order multiscale methods, we implement and solve the same problem on the same mesh in *FreeFem++* [81]. *FreeFem++* is a widely-used software written in C++ to solve partial differential equations in two and three dimensions. We solve $\boldsymbol{\Phi}_{E,1}$ and $\pi_{E,1}$ of local problems (4.37)–(4.43) associated to the boundary E on the mesh shown in Figure 4.10.

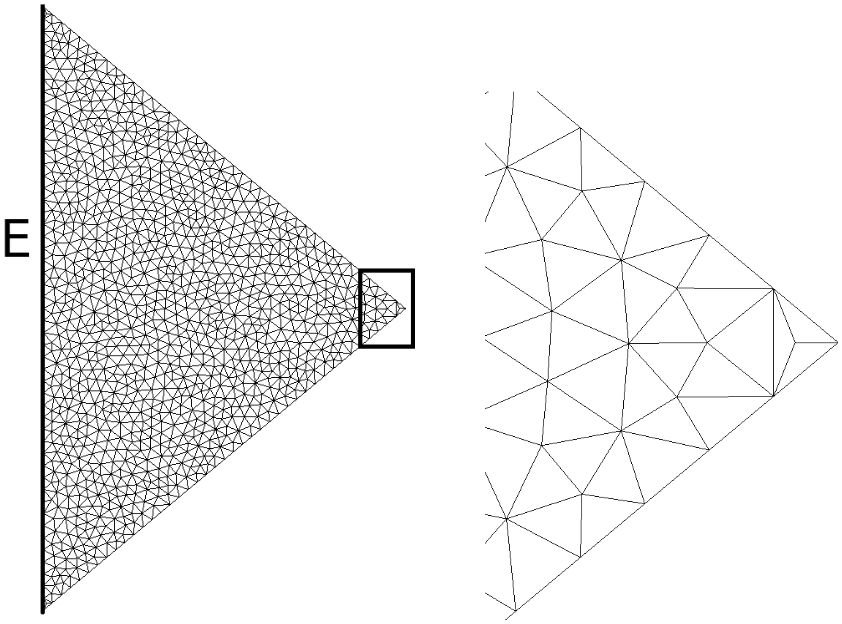


FIGURE 4.10: Mesh on the computational domain and zoom of a corner

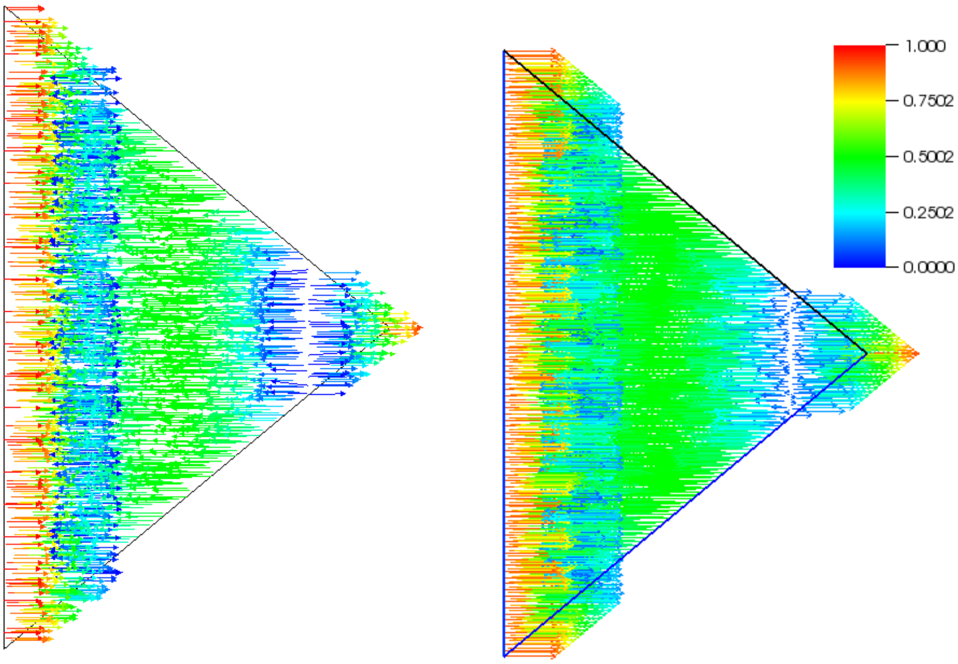


FIGURE 4.11: Velocity $\Phi_{E,1}$ computed by *TrioCFD* (left) and *FreeFem++* (right)

We first discretize local problems (4.37)–(4.43) with the Crouzeix-Raviart finite element. We observe that there are almost no differences between the velocity and pressure fields computed by *TrioCFD* and by *FreeFem++*. Then we discretize the local problem (4.37)–(4.43) with the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element and solve (4.57) with *TrioCFD* and *FreeFem++*. In *TrioCFD*, when the pressure is discretized in the $\mathbb{P}_1$ space, it is recommended to use a function *VerifierCoin*. As shown in Figure 4.10, this function allows to cut an element consisting of two faces on the boundary Γ_a into three elements. This operation is not performed in *FreeFem++*.

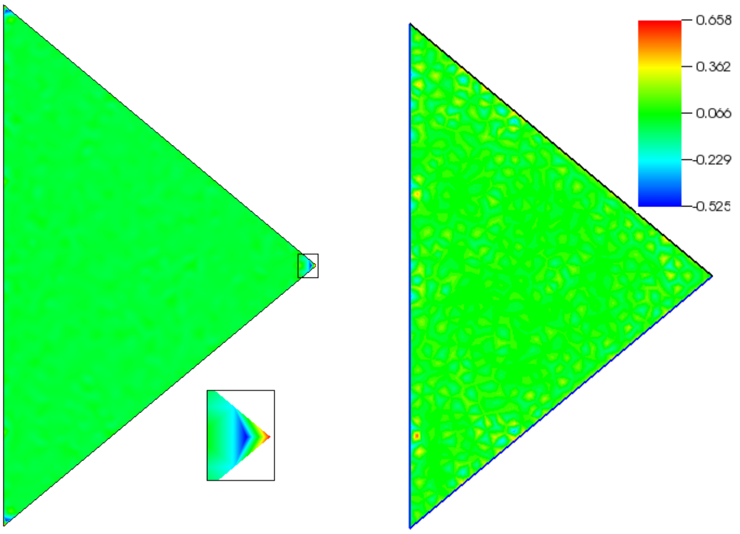
Denoting by h_f the average mesh size, we first solve the local problem on a mesh with $h_f = 0.02$. As shown in Figure 4.11, there are no significant differences between the velocity field given by *TrioCFD* and *FreeFem++*. However, as shown in Figure 4.12 (a), the pressure field given by these software is almost the same except some very slight differences. In the pressure field given by *TrioCFD*, we observe that there are some strong extrema localized in the corners of the domain.

In order to understand this phenomenon, we refine the mesh by taking $h_f = 0.01$ and $h_f = 0.005$. As shown in Figure 4.12 (b) and (c), we observe that the magnitude of the pressure field computed by *FreeFem++* decreases and approaches zero whereas the extrema in the pressure field computed by *TrioCFD* do not decrease at all. Then we solve $\mathbf{\Phi}_{F,3}$ and $\pi_{F,3}$ of (4.37)–(4.43) in the domain shown in Figure 4.13. As shown in Figure 4.14, in the pressure field $\pi_{F,3}$ computed by *TrioCFD*, we still observe some strong extrema localized in the corners of the computational domain. This phenomenon is not observed in the pressure field computed by *FreeFem++*.

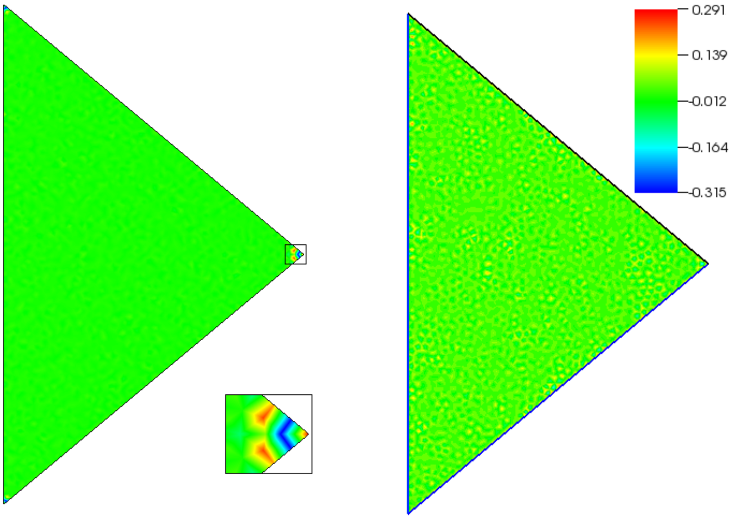
Using the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element, our investigations reveal that the pressure field computed by *TrioCFD* is a little disappointing. This was probably related to the intrinsic implementation of the $\mathbb{P}_1$ discretization of the pressure in the software, for example, the treatment of boundary conditions. The reasons for this phenomenon are not entirely understood during this thesis. Despite the existence of localized extrema in the pressure field, we can conclude nevertheless that the pressure and the velocity computed by *TrioCFD* are in good agreement with those computed by *FreeFem++*.

As presented in Chapter 3, velocity solutions of local problems form a basis of the velocity approximation space V_H. These multiscale basis functions are involved in the computation of matrices on coarse elements and thus in the solution of coarse-scale problems. Since velocity solutions of local problems are correctly computed with the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element, it is understandable that high-order Crouzeix-Raviart multiscale methods can still be applied to solve flow problems. As defined by (3.208), pressure solutions of local problems are only used in the reconstruction of the fine-scale pressure.

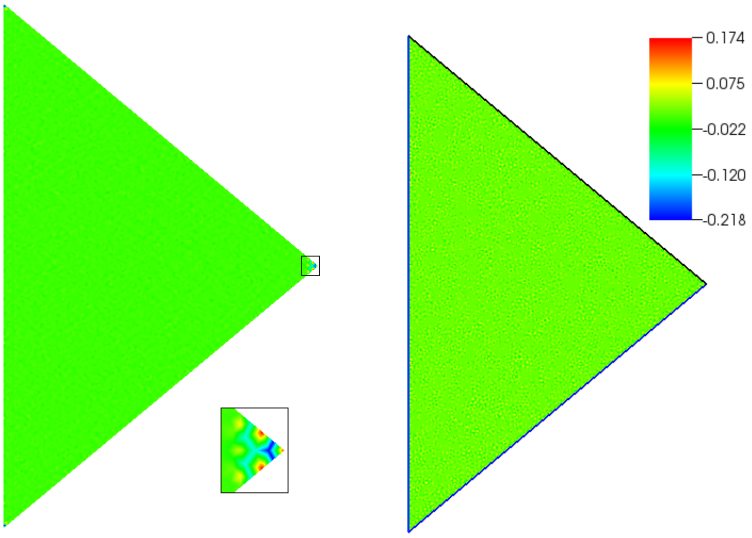
Our numerical results (see Chapter 6) highlighted that high-order Crouzeix-Raviart multiscale methods outperform all other Crouzeix-Raviart multiscale methods presented in this thesis. In particular, when local problems are discretized using the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element, high-order Crouzeix-Raviart multiscale methods yield much more accurate pressure solutions than other multiscale methods. This confirms that the slight problem in the pressure solution of local problems is of limited consequence to the accuracy of the fine-scale pressure computed by high-order multiscale methods.



(a) $h_f = 0.02$



(b) $h_f = 0.01$



(c) $h_f = 0.005$

FIGURE 4.12: Pressure $\pi_{E,1}$ computed by *TrioCFD* (left) and *FreeFem++* (right)

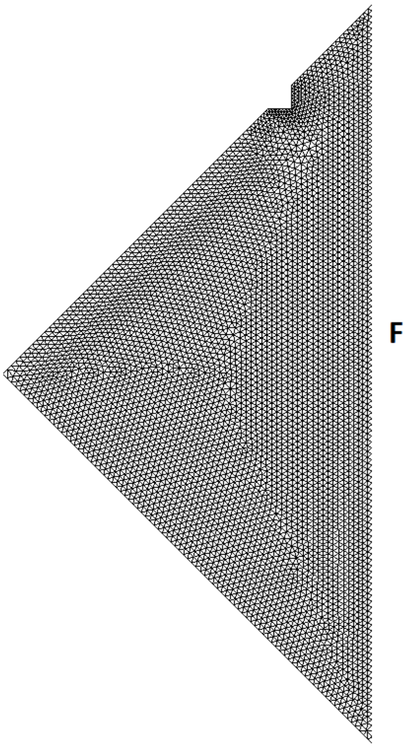


FIGURE 4.13: Mesh on the computational domain with an obstacle

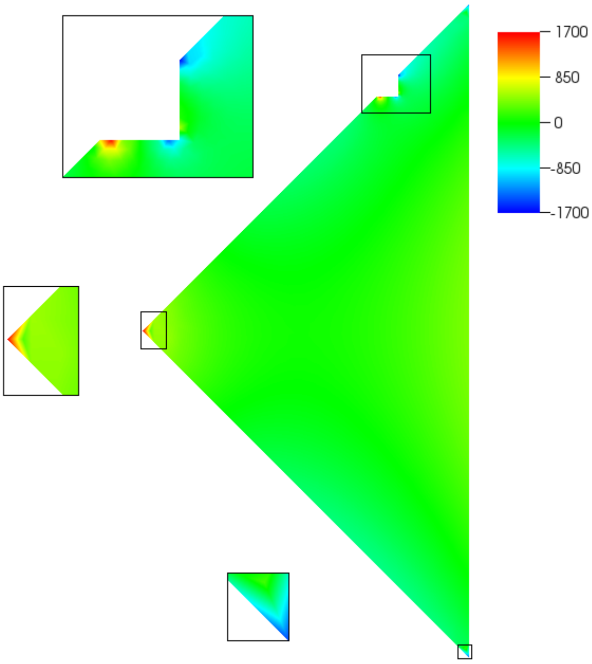


FIGURE 4.14: Pressure $\pi_{F,3}$ computed by *TrioCFD* in the domain with an obstacle

Chapter 5

The multiscale simulation chain

In this chapter, we present the multiscale simulation chain *SALOME-TrioCFD-VisIt* developed in the thesis. The multiscale simulation chain consists of all necessary tools and preliminary preparations for numerical simulations with multiscale finite element methods. The simulation chain consists of three main tools: the *SALOME* platform [129], the solver *TrioCFD* [139] and the visualization tool *VisIt* [142]. Note that there exist two types of parallelisms in the multiscale simulaton chain, the intrinsic extra-cellular parallelism and the intra-cellular parallelism. Both parallelisms allow to reduce computing time and memory.

The *SALOME* platform is used in the first step of the simulation chain to prepare mesh files necessary for multiscale finite element methods. Since the generation of such meshes is not a native function of *SALOME*, an important work was devoted to the development of some specific algorithms in *SALOME*. The solver *TrioCFD* is involved in the second step of the multiscale simulation chain to perform numerical simulations. The main steps of multiscale finite element methods are carried out in *TrioCFD*, including the solution of local problems, the assembly and the solution of coarse-scale problems as well as the reconstruction of fine-scale solutions. These operations are particularly related to multiscale finite element methods and they are not available in *TrioCFD*. A considerable work was devoted to the implementation of these operations *TrioCFD*. At the end of the multiscale simulation chain, *VisIt* is used to visualize the reconstructed fine-scale solutions.

Outline Section 5.1 presents the main steps of the multiscale simulation chain as well as the parallelism of each step. Section 5.2 presents the *SALOME* platform and algorithms developed for the generation of meshes. Section 5.3 presents *TrioCFD* and important developments related to multiscale methods. Section 5.4 discusses briefly the visualization of solutions in *VisIt*.

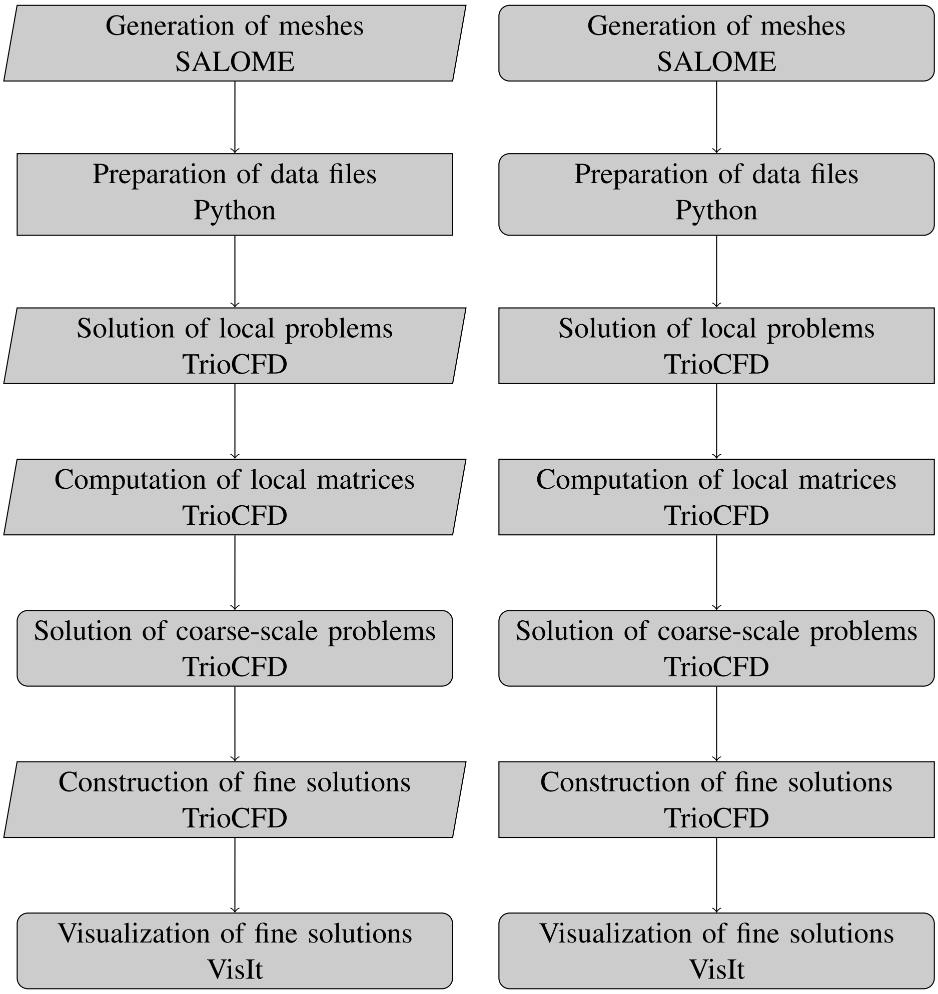
5.1 Parallelisms in the simulation chain

The main steps of the *SALOME-TrioCFD-VisIt* multiscale simulation chain are illustrated in Figure 5.1. The theoretical and the practical parallelisms of each step are represented in different colors. We distinguish two types of parallelism: extra-cellular parallelism and intra-cellular parallelism. Both parallelisms contribute to the reduction of computing time and memory.

The extra-cellular parallelism is an intrinsic parallelism of multiscale finite element methods. As presented in Chapter 3, Crouzeix-Raviart multiscale finite element methods are nonconforming methods, i.e. only the average jump of velocity is required to be continuous along the interfaces of coarse elements. Local problems are solved on each coarse element independently from adjacent ones. Similarly, the computation of matrices and the reconstruction of fine-scale solutions are also carried out in each coarse element without communication with neighboring ones. Therefore these operations can be carried out in parallel by executing one instance of *TrioCFD* on each coarse element.

The intra-cellular parallelism allows to assign several processors for one task. *SALOME* has parallelized meshing tools which permit to generate large mesh files using several processors in order to reduce computing time. Some solvers in *TrioCFD* are parallelized in order to solve large systems with numerous processors. In parallel computations, *TrioCFD* has an optimal performance provided that each processor treats about 20,000-30,000 mesh elements. This rule helps the user to decide the number of processors to use in the computation.

Figure 5.1 (a) and (b) show respectively the theoretical and implemented parallelism in the multiscale simulation chain. Theoretically, all expensive steps (shown in green), such as the generation of meshes and the solution of local problems have both the extra-cellular parallelism and intra-cellular parallelism. However, in practice, the intra-cellular parallelism was not developed for these steps for two main reasons. The first reason is that the intra-cellular parallelism is not necessary for these steps. For example, the computation of matrices and the reconstruction of fine-scale solutions are both very fast even with only one processor. The solution of local problems (with the *prediction-correction* algorithm) is relatively efficient sequentially if fine meshes contain less than 30,000 elements. The second reason is a lack of time. In fact, in three dimensions, some coarse elements can contain more than 100,000 elements due to the complexity of the geometry. When solving local problems on these coarse elements, it is desired to use several processors in order to reduce computing time.



(a) Theoretical parallelisms (b) Implemented parallelisms

FIGURE 5.1: Main steps of the multiscale simulation chain and associated parallelisms: sequential computation (rounded corner rectangle), the extra-parallelism (rectangle) and the intra- and extra-parallelism (trapezium)

5.2 Pre-processings in the SALOME platform

As presented in subsection 2.5.1 of Chapter 2, multiscale finite element methods consist of two types of meshes: the coarse mesh $\mathcal{T}_H$ and a fine mesh $\mathcal{T}_h(K)$ on each coarse element $K \in \mathcal{T}_H$. The coarse mesh $\mathcal{T}_H$ contains no obstacles whereas the mesh $\mathcal{T}_h(K)$ should be fine enough to resolve the boundary of obstacles in the coarse element. The coarse-scale mesh and fine meshes are particularly related to multiscale methods and thus require some special developments in *SALOME*. In this section, we present the *SALOME* platform and the main algorithms developed for the generation of meshes in this thesis.

5.2.1 The *SALOME* platform

SALOME [129] is an integration platform developed by CEA and EDF. It is distributed as open-source software under the terms of the GNU LGPL license. *SALOME* provides a generic platform for Pre- and Post-Processings for numerical simulations using spatially discretized, mesh-based methods like finite element or finite volume methods. The platform is based on an open and flexible architecture made of reusable components. Besides, it is parallelized for the generation of very complex or large geometries and meshes.

The platform contains several separate working modes for geometry creation or manipulation (*GEOM*), meshing (*MESH*), post-processing (*ParaVIS*) and manipulation of physical fields based on meshes (*FIELDS*). The *GEOM* module has quite extensive capabilities in creation and manipulation of geometries. The geometry management is based on Open CASCADE Technology (OCCT) [119] and Visualization Toolkit (VTK) visualization library.

The MESH module is capable to generate various two-dimensional (2D) and three-dimensional (3D) meshes, for example, 3D meshes consisting of tetrahedra or hexahedra or a mix of tetrahedra and hexahedra. For 2D surface meshing, we can use tools like NETGEN, Mefisto and MeshGems-CADSURF etc. For 3D volume meshing, there are various tools such as NETGEN, MeshGems-Tetra, MeshGems-Hexa, MeshGems-Hybrid, Gmsh and so on. MeshGems-CADSURF and MeshGems-Tetra are commercial meshers developed by Distene [55]. Mesh files can be exported in the MED format which is the standard and default mesh format of *SALOME*. Moreover, if necessary, mesh files can also be exported in other formats such as UNV, DAT and so on. Using the *ParaVIS* module, *SALOME* can do post-processings of numerical results and do some manipulations on meshes such as merging and splitting.

In *SALOME*, we can access to almost all existing functions via the Graphic User Interface (GUI). We can also use the TUI (Text User Interface) which provides a functionality to access to all features from the Python console (embedded in GUI desktop or an external one). The TUI is more practical for the creation of complicated geometries and meshes. It allows to change geometric or mesh parameters easily for the generation of multiple geometries and meshes. In this thesis, we used the Python console for the generation of all geometries and meshes. We used the version *SALOME-8.2.0* to export mesh files that can be manipulated by *TRUST-1.7.5*. We will present *TRUST-1.7.5* later in this chapter. Since the development of *SALOME* is very active, several updated versions have come out during this thesis, but all functions used in this thesis have not been impacted.

In this thesis, we have mainly used commercial meshers MeshGems-CADSURF and MeshGems-Tetra [55] respectively for 2D and 3D meshing under the license of CEA. All mesh files are exported in the MED format which can be manipulated easily by *TrioCFD*.

5.2.2 The GEOM and MESH modules

In the upstream preparation for numerical simulations, GEOM and MESH modules are used respectively to generate the geometry and the associated mesh.

In a first step, the GEOM module is used to create a geometric object which represents the computational domain. In this object, each characteristic zone such as the boundary of the domain is represented as a geometric group. Each geometric group can contain several objects but it represents only one physical property. The TUI allows to get access to all existing functions in GEOM. Making use of these functions, we have developed algorithms in Python for the generation of complicated geometries. These algorithms allows us to automate the generation of geometries and to change easily geometric parameters.

In a second step, the MESH module is used to discretize the geometric object into finite elements such as triangles, tetrahedra and so on. A meshing tool is executed to get the mesh object associated to the geometry object. Mesh groups are created from geometric groups created in the first step. Mesh groups contain boundary elements which are necessary for the treatment of boundary conditions in numerical simulations. At the end of this step, the mesh object and groups are exported into a mesh file in the MED format.

It should be noted that geometric objects created in the first step is essential for normal operations on the mesh object. If we delete the geometric object, all operations on the mesh object become impossible except the display of meshes. In practice, it takes more work to create geometric objects than to create mesh objects.

In this thesis, we work on a perforated domain Ω^ε which is rectangular ($d = 2$) or parallelepipedic ($d = 3$). The discretization of the domain is already briefly presented in subsection 2.5.1 and we recall some notations here. As shown in Figure 3.1, both the coarse mesh and fine meshes are made of either triangles ($d = 2$) or tetrahedrons ($d = 3$). Moreover, we denote by $\mathcal{T}_H(\Omega)$ the coarse mesh of the domain Ω and denote by $\mathcal{T}_h(K)$ the fine mesh on the coarse element $K \in \mathcal{T}_H(\Omega)$. For the sake of simplicity, for any $K \in \mathcal{T}_H(\Omega)$, we denote by B_K^ε the set of obstacles intersecting with K, i.e. $B_K^\varepsilon = B^\varepsilon \cap K \neq \varnothing$. Besides, we define $K^\varepsilon = K \setminus \bar{B}_K^\varepsilon$ and $E^\varepsilon = E \setminus \bar{B}_K^\varepsilon$ for any $E \in \partial K$.

5.2.3 Parallelisms of the generation of meshes

As shown in Figure 5.1 (a), theoretically, the generation of meshes benefits from both the intra-cellular and extra-cellular parallelisms. Let us discuss first the extra-cellular parallelism. As presented in Chapter 3, Crouzeix-Raviart multiscale finite element methods are nonconforming methods, i.e. only the average jump of velocity is required to be continuous along the interfaces of coarse elements. Local problems are solved on each coarse element independently from adjacent ones. Similarly, the reconstruction of fine-scale solutions is also carried out in each coarse element without communication with adjacent ones. Thus Crouzeix-Raviart multiscale finite element methods allow to use fine meshes that are non-matching along the interfaces of coarse elements.

Without the constraint of conformity, non-matching meshes provide more flexibility when meshing complicated geometries. Theoretically, non-matching fine meshes meet the need of Crouzeix-Raviart multiscale methods and they can be generated in parallel by executing one instance of *SALOME* on each coarse element. However, since the generation of fine meshes is very fast, we execute in practice only one instance of *SALOME* and create the fine mesh one after another.

Now we discuss the intra-cellular parallelism. Theoretically, the generation of both the coarse and fine meshes can benefit from the intra-cellular parallelism. In other words, it is possible to assign several processors when using parallelized meshing tools in *SALOME*. In practice, since the generation of fine meshes is efficient even sequentially, we have not parallelized the algorithms developed in this thesis. As a result, the generation of meshes is marked as a sequential step in Figure 5.1 (b).

In order to validate Crouzeix-Raviart multiscale methods, we also solve the multiscale problem on a reference mesh with the traditional numerical method in *TrioCFD* and this solution is called the reference solution. As shown in Figure 3.1 of Chapter 3, the reference mesh is the union of fine meshes on coarse elements provided that fine meshes are matching along interfaces of coarse elements. Due to the constraint of conformity, it is more complicated to create conforming fine meshes than nonconforming ones in *SALOME*. Note that the reference mesh is not required by multiscale methods and it is only needed for validation purpose during the development phase of Crouzeix-Raviart multiscale methods.

In this thesis, we distinguish two types of heterogeneous media: (a) with randomly placed obstacles; (b) with periodically placed obstacles. For the media (a), we have developed an algorithm for the generation of conforming fine meshes (see subsection 5.2.4) and an algorithm for the generation of nonconforming fine meshes (see subsection 5.2.5). For the media (b), we have developed a special algorithm which takes advantage of the periodicity of the media (see subsection 5.2.7).

5.2.4 Generation of the coarse mesh and conforming fine meshes

The generation of the coarse mesh and conforming fine meshes is described in Algorithm 5.1. The conformity of fine meshes along the interfaces of coarse elements is guaranteed using the fine mesh $\mathcal{T}_h(\Omega)$. This algorithm is described in a simplified Python script provided in Appendix A.1.

Algorithm 5.1 Generation of the coarse mesh and conforming fine meshes

```
1: Discretize the domain $\Omega$ into a coarse mesh $\mathcal{T}_H(\Omega)$      ▷ subsection 5.2.4.1
2: Create the set of obstacles $B^\varepsilon$                                  ▷ subsection 5.2.4.2
3: Discretize the domain $\Omega^\varepsilon$ into a fine mesh $\mathcal{T}_h(\Omega^\varepsilon)$   ▷ subsection 5.2.4.3
4: for each element $K \in \mathcal{T}_H(\Omega)$ do
5:     Create the fine mesh $\mathcal{T}_h(K)$ based on $\mathcal{T}_h(\Omega^\varepsilon)$        ▷ subsection 5.2.4.3
6:     Export the fine mesh $\mathcal{T}_h(K)$ into a MED file
7: end for
```

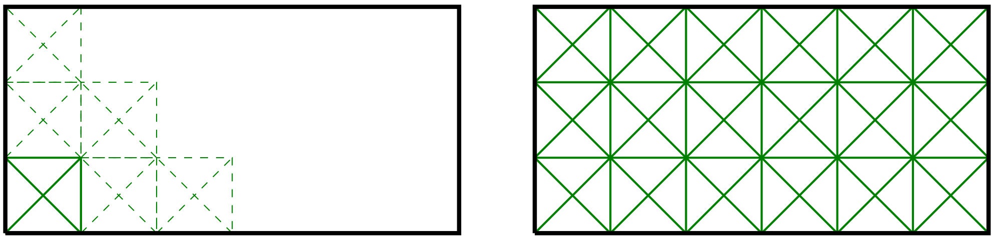
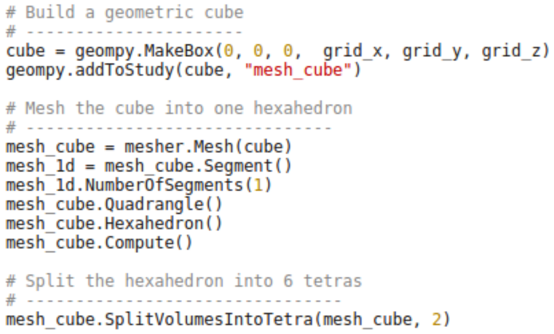

(a) domain Ω (b) coarse mesh $\mathcal{T}_H(\Omega)$

FIGURE 5.2: An illustration of the generation of the coarse mesh

5.2.4.1 Generation of the coarse mesh

The coarse mesh consists of a connected union of coarse elements which can be of any shape theoretically. It was pointed out in [61] that an appropriate choice of the coarse mesh will improve the efficiency and accuracy of multiscale approaches. In this thesis, the coarse mesh consists of either triangular elements ($d = 2$) or tetrahedral elements ($d = 3$).

An shown in Figure 5.2 (a), we first create a square ($d = 2$) or a cubic ($d = 3$) geometric object in the GEOM module. Then we transform the geometric object into a mesh object which access to functions of the MESH module. In the MESH module, we use the function **QuadTo4Tri** ($d = 2$) to split the square into 2 or 4 triangles or **SplitVolumesIntoTetra** ($d = 3$)



```
# Build a geometric cube
# ----------------------
cube = geompy.MakeBox(0, 0, 0,  grid_x, grid_y, grid_z)
geompy.addToStudy(cube, "mesh_cube")

# Mesh the cube into one hexahedron
# -------------------------------
mesh_cube = mesher.Mesh(cube)
mesh_1d = mesh_cube.Segment()
mesh_1d.NumberOfSegments(1)
mesh_cube.Quadrangle()
mesh_cube.Hexahedron()
mesh_cube.Compute()

# Split the hexahedron into 6 tetras
# --------------------------------
mesh_cube.SplitVolumesIntoTetra(mesh_cube, 2)
```

FIGURE 5.3: An example of the function **SplitVolumesIntoTetra**

to split the cube into 6 or 24 tetrahedrons. The Python script shown in Figure 5.3 reveals the function **SplitVolumesIntoTetra**.

In a second step, we transform each element in the mesh object into a geometric object. Note that geometric objects are easier to manipulate than mesh objects in *SALOME*. As shown in Figure 5.2 (a), we translate the geometric object to form a set of triangles ($d = 2$) or tetrahedrons ($d = 3$) that cover the whole domain Ω. Making use of the function **MakePartition** or **MakeGlueFaces**, the union of these geometric objects forms a unique geometric object Ω which consists of triangular or tetrahedral coarse elements. Then we create geometric groups on this geometric object, such as groups of boundaries of Ω.

In a third step, we transform the geometric object to a mesh object which is the coarse mesh $\mathcal{T}_H(\Omega)$ shown in Figure 5.2 (b). Based on the geometric groups created in the last step, we create mesh groups on the mesh object and the mesh groups serve to impose boundary conditions in numerical simulations. At last, the coarse mesh $\mathcal{T}_H(\Omega)$ is exported into a MED file, which contains the discretization (elements and connectivities) and mesh groups.

5.2.4.2 Creation of obstacles B^ε

Given a list of coordinates and sizes of obstacles in the domain, it is straightforward to create geometric objects of obstacles in the GEOM module. If obstacles are all of the same size, we can create one geometric object and translate it elsewhere using the function **MakeTranslation**. At last, the function **MakeCompound** assembles all geometric objects of obstacles to form one unique geometric object B^ε.

5.2.4.3 Generation of $\mathcal{T}_h(\Omega^\varepsilon)$ and conforming fine meshes

In the GEOM module, taking the geometric object Ω as the source object and the geometric object B^ε as the tool object, the boolean operation **MakeCut** creates the perforated domain Ω^ε. As described in subsection 5.2.4.1, the geometric object Ω is an union of geometric objects of elements $K \in \mathcal{T}_H(\Omega)$. Thus after using **MakeCut**, the geometric object Ω^ε is an union of geometric objects of elements $K^\varepsilon = K \setminus \bar{B}^\varepsilon_K$ for $K \in \mathcal{T}_H(\Omega)$. On the geometric object Ω^ε, we create a geometric group of faces E^ε for $E \in \mathcal{E}_H$ and a group of solids K^ε for $K \in \mathcal{T}_H(\Omega)$.

In the MESH module, we create a surface mesh on the group of faces using **MG_CADSurf**. This operation determines the position and the number of meshing nodes on each face E^ε for $E \in \mathcal{E}_H$. Then under the constraint of the surface mesh, we create a volume mesh $\mathcal{T}_h(K)$ using **MG_Tetra** on each solid K^ε for $K \in \mathcal{T}_H(\Omega)$. At the end, **MG_Tetra** takes automatically the union of volume meshes and produces a unique fine mesh $\mathcal{T}_h(\Omega^\varepsilon)$. The volume mesh generated on each solid is exported into a fine mesh $\mathcal{T}_h(K)$ for $K \in \mathcal{T}_H(\Omega)$.

At last, we create mesh groups respectively on $\mathcal{T}_h(\Omega^\varepsilon)$ and on each fine mesh $\mathcal{T}_h(K)$ for $K \in \mathcal{T}_H(\Omega)$. All the mesh objects are exported into MED files which contain all necessary information for numerical simulations, including the discretization (elements and connectivities) and mesh groups for boundary conditions.

By construction, neighboring fine meshes share the same meshing nodes on their interface and we obtain conforming fine meshes. Proceeding in this way, the generation of $\mathcal{T}_h\left(\Omega^\varepsilon\right)$ in very efficient since the mesh is generated by taking the union of small fine meshes which are very fast to create.

5.2.5 Generation of the coarse mesh and nonconforming fine meshes

As mentioned previously, the reference mesh is not required in multiscale finite element methods and it is needed only for validation purpose. After the validation phase, the reference mesh is not required and it is better to create non-matching fine meshes in multiscale simulations. Now we present the generation of nonconforming meshes following Algorithm 5.2.

Algorithm 5.2 Generation of the coarse mesh and nonconforming fine meshes

```
Discretize $\Omega$ into a coarse mesh $\mathcal{T}_H(\Omega)$                ▷ subsection 5.2.5.1
for each element $K$ in the coarse mesh $\mathcal{T}_H(\Omega)$ do
    Create the set of obstacles $B_K^\varepsilon$                            ▷ subsection 5.2.5.2
    Assemble the macroelement $K^\varepsilon = K \setminus \bar{B}_K^\varepsilon$              ▷ subsection 5.2.5.3
    Discretize the macroelement $K^\varepsilon$ to a fine mesh $T_h(K)$       ▷ subsection 5.2.5.4
    Export the mesh $\mathcal{T}_h(K)$ into a MED file
end for
```

5.2.5.1 Generation of the coarse mesh

The generation of the coarse mesh is the same as what was presented in subsection 5.2.4.1.

5.2.5.2 Creation of obstacles B_K^ε

Given a list of coordinates and sizes of obstacles in the domain, for each coarse element $K \in \mathcal{T}_H$, we need to determine the set of obstacles which intersect with K. In this thesis, we take the distance between the gravity center of K and that of the obstacle as the characteristic length. Since obstacles are either circular or square for $d = 2$ and cylindrical for $d = 3$, it is easy to check with existing functions of *SALOME* whether an obstacle intersects with K.

5.2.5.3 Assembly of the coarse element K^ε

For each $K \in \mathcal{T}_H\left(\Omega\right)$, the coarse element $K^\varepsilon = K \setminus \bar{B}_K^\varepsilon$ is created easily by the function **MakeCut** in the GEOM module. It is a boolean operation which takes the element K as the source object and obstacles B_K^ε as the tool object. It returns the source object cut by obstacles, i.e. the coarse element K^ε. Then on the coarse element K^ε, we create geometric groups of boundaries of K^ε and a geometric group of boundaries of B_K^ε.

5.2.5.4 Generation of nonconforming fine meshes

In the MESH module, we discretize each geometric object K^ε independently from neighboring objects and obtain the fine mesh $\mathcal{T}_h\left(K\right)$. Thus the conformity on the interface of adjacent fine meshes is not guaranteed and thus the union of fine meshes is nonconforming. The surface and volume meshes are generated respectively by **MG_CADSurf** and **MG_Tetra**. On each fine mesh, we create mesh groups based on the geometric groups created in the last step. At last, each fine mesh is exported into a MED file which contains the discretization (elements and connectivities) and mesh groups.

5.2.6 Treatment of tangent points

In Algorithm 5.1 and Algorithm 5.2, for $K \in \mathcal{T}_H$, when creating K^ε using B^ε and K, randomly placed obstacles may be tangent to ∂K. Thus the intersection of B^ε and K is reduced to only isolated points and the function **MakeCut** in the GEOM module might fail to generate K^ε. Even though **MakeCut** is able to create K^ε, meshing tools in the MESH module probably fail due to isolated tangent points. When obstacles are not strictly tangent but very close to ∂K, a very fine mesh is needed in order to resolve the tiny space between obstacles and ∂K. Moreover, it is possible that *TrioCFD* can not produce correct solutions on such kind of meshes.

However, tangent points are difficult to avoid especially when there is a large number of obstacles. One solution to this problem is to create a coarse mesh consisting of polygons. The shape of polygons can be designed to avoid tangent points with obstacles. Note that theoretically, coarse elements of the coarse mesh can be of any shape. However, *TrioCFD* can not read this kind of meshes for the moment.

Another solution is to discretize obstacles and use the penalization technique [21] to impose the non-slip boundary condition on obstacles. However, in *TrioCFD*, it is recommended to use body-fitted unstructured meshes in numerical simulations. Obstacles are not discretized and no penalization technique is used. In periodic heterogeneous domain, tangent points can be avoided more easily by choosing an appropriate coarse mesh according to the distribution of obstacles.

5.2.7 A special algorithm for periodic heterogeneous media

The case of flow through closely spaced periodic cylinder arrays is important for nuclear industrial applications, e.g. flow around nuclear fuel rods, flow past heat exchange coils in steam generators and so on. A medium with closely spaced periodic cylinder arrays is a typical example of periodic heterogeneous media.

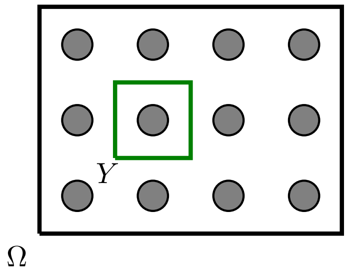


FIGURE 5.4: Illustration of a periodic domain

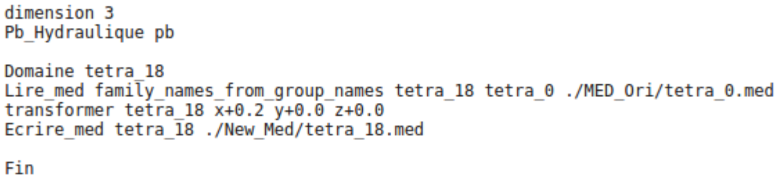
Algorithm 5.3 A special algorithm for periodic media

```
1: Generate the coarse mesh $\mathcal{T}_H(Y)$ and fine meshes $\mathcal{T}_h(K)$ for $K \in \mathcal{T}_H(Y)$ using Algorithm 5.2
2: Solve local problems on each fine mesh $\mathcal{T}_h(K)$
3: Compute matrices locally on each fine mesh $\mathcal{T}_h(K)$
4: Translate each fine mesh $\mathcal{T}_h(K)$ elsewhere to obtain new meshes $\mathcal{T}_h(K')$
5: Copy basis functions and matrices computed on each $\mathcal{T}_h(K)$ to new meshes $\mathcal{T}_h(K')$
```

As shown in Figure 5.4, a periodic domain can be considered as the translation of a reference cell Y. We remark that in Crouzeix-Raviart multiscale finite element methods, basis functions and matrices computed on Y are the same as those computed on other cells. If the periodic domain Ω^ε contains an important number of small obstacles, it can take a lot of computing time to generate fine meshes following Algorithm 5.2. Besides, it can be expensive to solve local problems on each fine mesh, especially in 3D (see Chapter 5). To reduce the computing time, we take advantage of the periodicity of the domain by using Algorithm 5.3. This algorithm presents the generation of fine meshes, the solution of local problems and the computation of matrices in periodic media.

In a first step, we create a simple geometric object Y in the GEOM module. Then following Algorithm 5.2, we create on the reference cell Y a coarse mesh $\mathcal{T}_H(Y)$ and fine meshes $\mathcal{T}_h(K)$ for $K \in \mathcal{T}_H(Y)$. The coarse mesh $\mathcal{T}_H(Y)$ consists of coarse elements, for example, 4 triangles if $d = 2$ or 24 tetrahedrons if $d = 3$.

In a second step, we solve local problems and compute matrices on each fine mesh $\mathcal{T}_h(K)$ for $K \in \mathcal{T}_H(Y)$. Since $\mathcal{T}_H(Y)$ consists of only several coarse elements, we need to solve only a small number of local problems and compute only several matrices. The solutions of local problems and the pre-computed matrices are stored for later use.



```
dimension 3
Pb_Hydraulique pb

Domaine tetra_18
Lire_med family_names_from_group_names tetra_18 tetra_0 ./MED_Ori/tetra_0.med
transformer tetra_18 x+0.2 y+0.0 z+0.0
Ecrire_med tetra_18 ./New_Med/tetra_18.med

Fin
```

FIGURE 5.5: A data file to execute the **transformer** function

In a third step, we use the function **transformer** of *TrioCFD* to translate fine meshes $\mathcal{T}_h(K)$ for $K \in \mathcal{T}_H(Y)$ in order to cover the whole domain Ω^ε. In the data file shown in Figure 5.5, we translate the source mesh tetra_0.med, following the translation vector $(0.2, 0, 0)$ and then obtain a new mesh named tetra_18.med. The generation of such data files are automated by executing a Python file. Numerical tests show that the translation is very fast and at the end of the translation, we obtain a set of fine meshes. Note that we tried also to translate fine meshes using some function in *SALOME* but it was more complicated.

Similarly, we copy basis functions and local matrices computed on fine meshes $\mathcal{T}_h(K)$ for $K \in \mathcal{T}_H(Y)$ to newly created fine meshes. This operation is automated by executing a Python script. Of course, we can also solve local problems and compute matrices on newly created fine meshes instead of using the copy operation.

5.3 Implementations in *TrioCFD*

In this section, we present practical implementations of Crouzeix-Raviart multiscale finite element methods in *TrioCFD*. We first introduce *TRUST* and *TrioCFD*. Then we present the implementations of the main steps of multiscale methods, including the solution of local problems, the computation of matrices, the solution of coarse-scale problems and the reconstruction of fine-scale solutions.

5.3.1 *TRUST* and *TrioCFD*

TrioCFD (previously named *Trio_U*) [139] is a free, open source CFD software, released and developed primarily by the Service de Thermohydraulics and Fluid Mechanics (STMF), Department of Nuclear Energy (DEN) in CEA. In 2015, *Trio_U* was separated into two parts: *TRUST* and *TrioCFD*, where *TRUST* (TRio_U Software for Thermohydraulics) is a new platform and *TrioCFD* is a *BALTIK* (Building Application Linked with TrioCFD Kernel) project based on *TRUST*. Figure 5.6 shows the *TRUST* platform and several *BALTIK* projects based on it.

TRUST [139] is developed in C++ (object-oriented) and it provides a general framework for the development of other projects and applications. In this framework, the structure of these

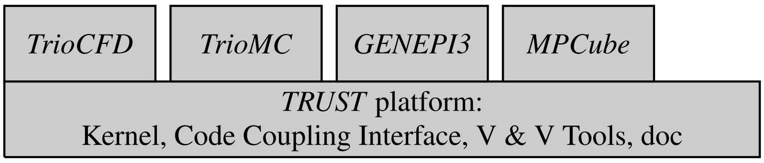


FIGURE 5.6: *Trust* platform and *BALTIK* projects

projects and applications is defined by some elementary objects: physical problems, equations, operators, time schemes, discretization and so on. Besides, *TRUST* provides an interface to some parallel numerical software libraries for partial differential equations and sparse matrix computations, such as PETSc [122]. *TRUST* also provides data structures and functions for the conception of parallel softwares, such as the management of memories and parallel I/O, distributed computations on vectors and matrices as well as distributed algorithms for the solution of linear systems. Other than *TrioCFD*, software developers can build other *BALTIK* projects based on the framework provided by *TRUST*, such as *TrioMC*, *GENEPI3* and *MPCube*.

Based on *TRUST*, *TrioCFD* has an extensive range of features to solve various thermo-hydraulics problems from turbulent flows to two-phase flows. It has also been successfully used to perform massive parallel calculations for nuclear safety studies. *TrioCFD* applies a hybrid finite volume based finite element method which is presented in Chapter 4. Velocity and temperature are discretized at the face center of triangular ($d = 2$) or tetrahedral elements ($d = 3$), leading to a $\mathbb{P}_1$ non-conforming discretization. The pressure can be discretized at the gravity center ($\mathbb{P}_0$), at the vertices ($\mathbb{P}_1$) or at both the gravity center and the vertices ($\mathbb{P}_0$ $\mathbb{P}_1$) of a finite element. The *prediction-correction* algorithm presented in section 4.3 is the only solver available to solve Navier-Stokes problems.

5.3.2 Preparations of data files for numerical simulations

In *TrioCFD*, numerical calculations are performed by executing data files, in which we specify simulation parameters, such as source terms, boundary conditions, discretization schemes, solvers and so on.

In Crouzeix-Raviart multiscale methods, we solve local problems locally on coarse elements and each local problem is specified in a data file. Besides, the assembly of matrices and the reconstruction of fine-scale solutions are performed on coarse elements. These operations need to be specified respectively in data files. It should be noted that these data files have been adapted for multiscale methods based on the native data file of *TrioCFD*. The working directories and data files are generated by executing a Python script developed in this thesis. As shown in Figure 5.1 (b), the preparation of data files is performed sequentially in practice. A commented data file for the solution of local problems is provided in Appendix A.2.

5.3.3 PROJECT_LOCAL_PB for local problems

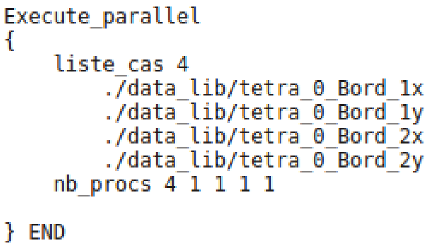
To solve local problems, we have created a new *BALTIK* project PROJECT_LOCAL_PB based on *TRUST-1.7.5*. In fact, the development of *TRUST* is very active and several updated versions have been released during this thesis. Since functions used in this thesis are not significantly affected, we prefer to continue the development using only one version of *TRUST*.

In PROJECT_LOCAL_PB, we have defined a new boundary condition named Cl_cellule to stand for the integral type of boundary condition. We have implemented local problems defined by Stokes equations (3.39) based on native Stokes equations implemented in *TrioCFD*. We have introduced some new unknowns i.e. Lagrange multipliers and implemented new matrices presented in (4.21).

Since Oseen equations are not natively available in *TrioCFD*, we have implemented local problems defined by Oseen problems (3.102) based on the original Navier-Stokes equations implemented in *TrioCFD*. We have mainly altered the nonlinear convection term to the linear Oseen term and modified the treatment of boundary conditions related to the Oseen term.

In an early stage of this thesis, we have solved local problems defined by Stokes equations (3.39) and Oseen equations (3.102) by modifying the native *prediction-correction* algorithm and this modification was already detailed in Chapter 4.

However, in a later stage of this thesis, for high-order multiscale methods, we have implemented local problems (3.191)–(3.192) completely differently from the native implementations of Stokes equations in *TrioCFD*. As already presented in section 4.6, an important work was devoted to the implementation of the matrix system (4.57) and a direct solver.



```
Execute_parallel
{
    liste_cas 4
        ./data_lib/tetra_0_Bord_1x
        ./data_lib/tetra_0_Bord_1y
        ./data_lib/tetra_0_Bord_2x
        ./data_lib/tetra_0_Bord_2y
    nb_procs 4 1 1 1 1

} END
```

FIGURE 5.7: An example of data file for **Execute_parallel**

Since most local problems are of small sizes, we have not developed the intra-cellular parallelism in PROJECT_LOCAL_PB. Each local problem is solved sequentially using only one processor. In three dimensions, local problems can be of large sizes and the intra-cellular parallelism might be desired. Since local problems are solved independently from one coarse element to another, the native function **Execute_parallel** in *TRUST* allows to benefit from this extra-cellular parallelism. As indicated by its name, this function permits to run multiple instances of *TRUST* to execute multiple data files at the same time. Besides, the number of processors is specified for each instance and can be different. As shown in Figure 5.7, **Execute_parallel** will solve four local problems by executing four instances of PROJECT_LOCAL_PB and one processor is assigned to each instance.

In multiscale methods, multiscale basis functions will be reused multiple times in coarse-scale problems and post-processings. *TRUST* can export solutions of local problems in several data formats, such as SAUV, LATA and MED. The SAUV format and the LATA format are the most used formats. In this thesis, multiscale basis functions are exported into the binary SAUV format which takes less storage space than other formats.

In order to validate the implementation of local problems in *TrioCFD*, we have solved the same problems using *FreeFem++* [81]. *FreeFem++* is a widely-used software written in C++ to solve partial differential equations. It can solve multiphysics nonlinear systems in two or three dimensions. In both *FreeFem++* and *TrioCFD*, we solve local problems (3.39) and (3.102) using the Crouzeix-Raviart element on the mesh shown in Figure 5.8. Note that weighting functions are chosen as indicated in the case $s = d$ of (3.7) and $d = 2$ in two dimensions.

Denoting by u_{triou} and u_{ff} the solution of *TrioCFD* and of *FreeFem++*, we define the relative difference by

$$d_r(u) = \frac{\|\boldsymbol{u}_{triou} - \boldsymbol{u}_{ff}\|_{L^2}}{\max(\|\boldsymbol{u}_{triou}\|_{L^2}, \|\boldsymbol{u}_{ff}\|_{L^2})}$$

Figure 5.9 shows the velocity field $\boldsymbol{\Phi}_{E,1}$ computed respectively by *Triou* and *Freefem++*. The boundary E is indicated in Figure 5.8. Table 5.1 confirms that there are no significant differences between solutions given by the two software. The difference of solutions of local

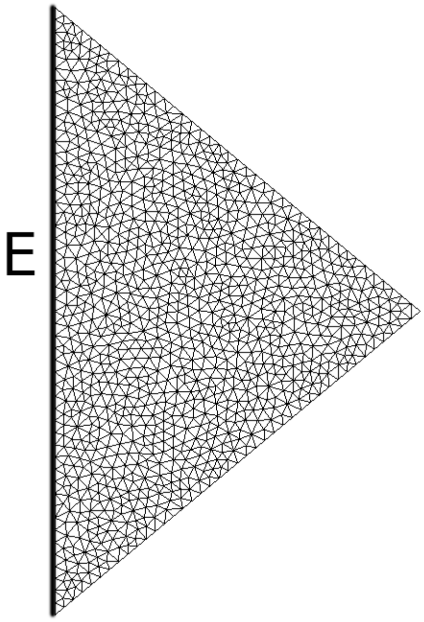
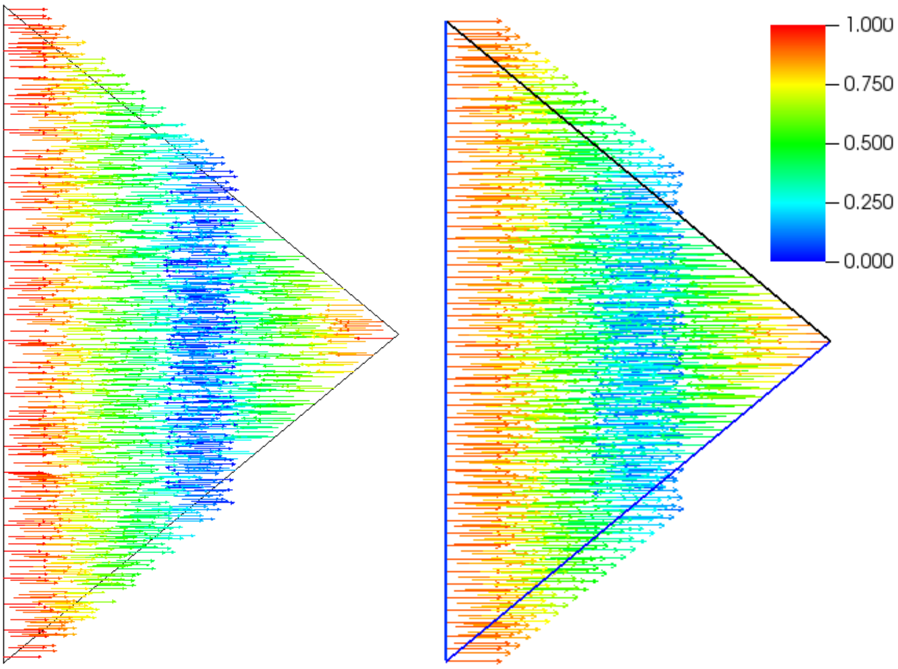


FIGURE 5.8: The fine mesh on a coarse element



(A) *TrioCFD*

(B) *FreeFem++*

FIGURE 5.9: Velocity $\mathbf{\Phi}_{E,1}$ computed by *TrioCFD* and of *FreeFem++*

TABLE 5.1: Error comparison of *Triou* and *Freefem++*

local problem	$d_r(u)$	$d_r(p)$
Stokes equations	6.193e-06	3.931e-05
Oseen equations	0.0168	0.018

problems defined by Oseen equations is due to the fact that the Oseen term is treated differently in the two software. In *TrioCFD*, the Oseen term is approximated by using the **upwind** scheme whereas the is not the case in *Freefem++*. We thus conclude that the implementation of local problems (3.39) and (3.102) in *TrioCFD* are valid.

5.3.4 PROJECT_MAT for matrix assembly

The coarse-scale problem is discretized by the finite element method specially developed to interface with multiscale basis functions. As presented in section 4.4, the coarse-scale problem is assembled from matrices locally computed on coarse elements. Thus we have created a *BALTIK* project PROJECT_MAT based on *TRUST* for the computation of matrices on coarse elements. On each coarse element, we read multiscale basis functions constructed on this element, compute matrices as presented in section 4.2 and then store matrices on the local disk.

These locally computed matrices will be used to assemble coarse-scale problems on the coarse mesh. For each element $K \in \mathcal{T}_H(\Omega)$ and each face $E \in \mathcal{E}_H$ in the coarse mesh, it is necessary to identify the related coarse element $K^\varepsilon = K \cap \Omega^\varepsilon$ and the related boundary $E^\varepsilon = E \cap \Omega^\varepsilon$. The objective is to find the right matrices computed locally on K^ε when assembling coarse-scale problems. Therefore, in PROJECT_MAT, we create a lookup table consisting of each boundary E^ε and the unit vector normal to E^ε. It is easy to see that the unit normal vector of the boundary E^ε and the face $E \in \mathcal{E}_H$ is the same. This table will be used for the identification of K^ε and E^ε when assembling coarse-scale problems.

5.3.5 PROJECT_COARSE_PB for coarse-scale problems

In multiscale finite element methods, coarse-scale problems are discretized by the finite element method presented in section 4.4. This finite element is specially developed to interface with multiscale methods and is not for independent use. The coarse-scale problems are implemented in a *BALTIK* project PROJECT_COARSE_PB created based on *TRUST*. In the general framework provided by *TRUST*, we have created a new discretization scheme *MsEF* and implemented elementary objects related to this discretization, such as *Operateurs*, *Champs*, *Zones* and so on.

Coarse-scale problems are assembled by reading pre-computed matrices stored on the local disk. For each element $K \in \mathcal{T}_H(\Omega)$ and each face $E \in \mathcal{E}_H$ in the coarse mesh, it is necessary to identify the related coarse element $K^\varepsilon = K \cap \Omega^\varepsilon$ and the boundary $E^\varepsilon = E \cap \Omega^\varepsilon$. The objective is to read the right matrices computed locally on K^ε and assemble correctly coarse-scale problems. The lookup table created in PROJECT_MAT is used to realize this objective.

Once the coarse-scale problem is assembled, we use existing solvers in *TrioCFD* to solve the probem. The only solver natively available is the *prediction-correction* algorithm. In the first stage of this thesis, we have used this algorithm to solve coarse-scale problems. Then in a later stage of this thesis, we have assembled the matrix system (4.60) and implemented a direct solver UMFPACK as presented in section 4.6. After the solution of coarse-scale problems, coarse solutions are stored on the local disk for the reconstruction of fine-scale solutions in the next step.

Since coarse-scale problems are assembled and solved on the coarse mesh, there is no extra-cellular parallelism in this step. Besides, coarse-scale problems are of limited sizes, with maximum hundreds of thousands unknowns, and can be solved efficiently sequentially. Thus the intra-cellular parallelism has not been developed in the thesis. In conclusion, coarse-scale problems are solved completely sequentially as shown in Figure 5.1 (b).

5.3.6 PROJECT_POS for the reconstruction of fine-scale solutions

The last step of the multiscale simulation chain is to reconstruct the fine-scale features of the velocity and pressure for visualization. This step was implemented in the *BALTIK* project PROJECT_POS created based on *TRUST*. Fine-scale solutions are reconstructed locally on each coarse element. We recall the reconstruction of fine-scale solutions presented in subsection 3.2.5: on each coarse element K^ε with $K \in \mathcal{T}_H(\Omega)$

$$\boldsymbol{u}_H = \sum_{E\in\mathcal{E}(K)} \sum_{i=1}^{s} u_{E,i} \boldsymbol{\Phi}_{E,i}$$

$$p_H = \sum_{E\in\mathcal{E}(K)} \sum_{i=1}^{s} u_{E,i} \pi_{E,i} + \bar{p}_H|_K$$

In practice, on the fine mesh $\mathcal{T}_h(K)$ associated to the coarse element K^ε, for $E \in \mathcal{E}(K)$ and $i = 1, \cdots, s$, we need to find from the local disk multiscale basis functions $\boldsymbol{\Phi}_{E,i}$ and $\pi_{E,i}$ that were constructed on this mesh. Then we need to find the value of $u_{E,i}$ from the coarse-scale velocity stored on the local disk. Similarly, we need to find the value of $\bar{p}_H|_K$ from the coarse-scale pressure $\bar{p}_H$ stored on the local disk. This identification process is a little complicated in practice and will not be detailed here. After finding the right $\boldsymbol{\Phi}_{E,i}$, $\pi_{E,i}$ and $u_{E,i}$ for $E \in \mathcal{E}(K)$ and $i = 1, \cdots, s$, we can construct the fine-scale velocity and pressure as defined by equations above. In this thesis, the re-construction of the fine-scale pressure is not emphasized.

At the end of this step, fine-scale solutions reconstructed on each coarse element are exported into a file in the LATA format. This format can be manipulated easily by *VisIt* which is the most used post-processings tool of *TrioCFD*. LATA files contain not only fine-scale solutions (velocity and pressure) but also the meshes used for the computation of solutions. Thus a large number of LATA files may require a large storage space. In the case of limited storage space, we can choose to reconstruct fine-scale solutions only in the regions of interest.

In Crouzeix-Raviart multiscale finite element methods, since fine-scale solutions are reconstructed locally on each coarse element, we manipulate each time only a reasonable amount of files that will not exceed modern computing capacities. Besides, we can choose to reconstruct fine-scale solutions on the entire domain or only in the regions of interest. This allows to reduce the total number of files to be manipulated and the required storage space.

To validate Crouzeix-Raviart multiscale methods, we compare reconstructed fine-scale solutions $\boldsymbol{u}_H$ and p_H with reference solutions $\boldsymbol{u}_{ref}$ and p_{ref}. The references solutions are computed by the classical numerical method implemented in *TrioCFD*. We defined the L^2 norm defined by

$$\|\boldsymbol{u}\|_{L^2} = \left(\sum_{K\in\mathcal{T}_H(\Omega)} \sum_{k\in\mathcal{T}_h(K)} \text{meas}(k) |\boldsymbol{u}(\boldsymbol{x}_k)|^2 \right)^{1/2}$$

where $\boldsymbol{x}_k$ represents the computing node of the physical unknown.

Using the L^2 norm, the relative errors of velocity $\boldsymbol{u}_H$ and of pressure p_H are defined by

$$L^2 \text{ Rel. U} = \frac{\|\boldsymbol{u}_H - \boldsymbol{u}_{ref}\|_{L^2}}{\|\boldsymbol{u}_{ref}\|_{L^2}}, \quad L^2 \text{ Rel. P} = \frac{\|p_H - p_{ref}\|_{L^2}}{\|p_{ref}\|_{L^2}} \tag{5.1}$$

In PROJECT_POS, after the reconstruction of fine-scale solutions, we compute on each coarse element K^ε the following term:

$$E_K^{loc}(\boldsymbol{u}_H) = \sum_{k\in\mathcal{T}_h(K)} \text{meas}(k) |(\boldsymbol{u}_H - \boldsymbol{u}_{ref})(\boldsymbol{x}_k)|^2$$

$$E_K^{loc}(\boldsymbol{u}_{ref}) = \sum_{k\in\mathcal{T}_h(K)} \text{meas}(k) |\boldsymbol{u}_{ref}(\boldsymbol{x}_k)|^2$$

By summing $E_K^{loc}(\boldsymbol{u}_H)$ and $E_K^{loc}(\boldsymbol{u}_{ref})$ on all coarse elements, we obtain $\|\boldsymbol{u}_H - \boldsymbol{u}_{ref}\|_{L^2}$

and $\|\boldsymbol{u}_{ref}\|_{L^2}$. Then it is straightforward to compute L^2 Rel. U defined by (5.1). The error L^2 Rel. P is computed in the same way.

Note that the computation of relative errors is not required in multiscale finite element methods. These errors are noly computed in the development phase of multiscale methods for validation purpose.

Remark. *In the SALOME platform, there is a module MEDCoupling which can compare numerical solutions computed on different meshes. Besides, MEDCoupling manipulates files in the MED format. However, during this thesis, TrioCFD is not able to export velocity located at element faces into the MED format. By interpolating the velocity computed at element faces, TrioCFD can obtain the velocity located at element centers or vertices and export it into the MED format. The interpolation introduces additional errors and we have no idea of the importance of these errors. Thus we did not use MEDCoupling to compute errors between reference solutions and reconstructed fine-scale solutions.*

5.4 Post-processings in *VisIt*

In this section, we present briefly the visualization of reconstructed fine-scale solutions using *VisIt*. In this thesis, we tried several methods and present only the most efficient one.

5.4.1 The visualization tool *VisIt*

VisIt [142] is one of the most-frequently used visualization tools. It is an open source, interactive, scalable, visualization, animation and analysis tool created by the Lawrence Livermore National Laboratory. Users can interactively visualize in parallel and analyze data ranging in scale from small (<101 core) desktop-sized projects to large (>105 core) leadership-class computing facility simulation campaigns. *VisIt* contains a rich set of visualization features to enable users to view a wide variety of data including scalar and vector fields defined on two- and three-dimensional structured, adaptive and unstructured meshes. Owing to its customizable plug-in design, *VisIt* is capable of visualizing data from various different scientific data formats. Besides, it includes a rich command line interface (CLI) based on Python which permits to launch *VisIt* by running Python scripts with several processors.

5.4.2 The visualization of fine-scale solutions in *VisIt*

At the end of the multiscale simulation chain, we use *VisIt* to visualize reconstructed fine-scale solutions. *VisIt* is the most used visualization tool of *TrioCFD* and it can manipulate easily LATA files produced by *TrioCFD*. Since the reconstruction of fine-scale solutions were carried out on each coarse element, we obtain as many LATA files as the number of coarse elements.

In this thesis, we execute *VisIt* by running a Python script where we specify some common parameters for all LATA files such as the legend, color tables, point of view and so on. A commented Python script for *VisIt* is presented in Appendix A.3. By running this script, *VisIt* reads one LATA file after another, visualize the desired physical field and export the visualization into a file in the PNG format. Finally, we obtain as many PNG files as the number of coarse elements. Then it is straightforward to glue the PNG files together to obtain one unique PNG file which shows the vision of the entire physical field or only the regions of interest. Besides, running this script using more processors can reduce the computing time. To summarize, in practice, the visualization of fine-scale solutions has no extra-parallelism but only the intra-parallelism as shown in Figure 5.1 (b).

Chapter 6

Numerical simulations

In this chapter, we present primarily applications of the multiscale simulation chain *SALOME-TrioCFD-VisIt* in two- and three-dimensional heterogeneous media. The heterogeneity is represented by the arbitrary or periodic placement of obstacles. We distinguish two types of media: (a) non-periodic heterogeneous media where obstacles are randomly placed; (b) periodic heterogeneous media where obstacles are periodically placed.

Throughout this chapter, the acronym MsFEMs stands for multiscale finite element methods. Fist of all, we apply the multiscale simulation chain to solve Stokes problems in a homogeneous medium using different Crouzeix-Raviart MsFEMs. The main objective is to validate the implementation of the multiscale simulation chain and to compare the accuracy of these MsFEMs.

In two-dimensional non-periodic heterogeneous media, we present numerical convergence studies and error analysis of various Crouzeix-Raviart MsFEMs presented in Chapter 3. These simulations are carried out for two objectives. The first objective is to show the numerical convergence of Crouzeix-Raviart MsFEMs defined by Stokes equations (see section 3.2) and by Oseen equations (see section 3.3). The second objective is to compare the accuracy of enriched multiscale methods, such as the multiscale method enriched by bubble functions (see section 3.5) and high-order Crouzeix-Raviart multiscale methods (see section 3.6). Three types of flow problems are concerned, including Stokes problems, Oseen problems and Navier-Stokes problems. In particular, we solve Navier-Stokes problems at different Reynolds numbers with the multiscale basis functions constructed respectively by Stokes equations or by Oseen equations.

In two-dimensional non-periodic heterogeneous media, we perform numerical convergence studies of various Crouzeix-Raviart MsFEMs presented in Chapter 3 and compare their accuracy. In three dimensions, Crouzeix-Raviart MsFEMs are applied to solve flow problems in non-periodic or periodic heterogeneous media. Due to limited computing capability, we are not able to compute reference solutions and thus no error analysis is made. Meanwhile, we present several demonstrative applications of Crouzeix-Raviart multiscale methods in highly heterogeneous media. The objective is to show their superior performance of MsFEMs compared to classical numerical methods in heterogeneous media with numerous solid obstacles.

Outline Section 6.1 describes notations used throughout this chapter. Section 6.2 presents applications of MsFEMs in a two-dimensional homogeneous medium. Section 6.3 presents theoretical results of the numerical convergence in the periodic case. Section 6.4 and section 6.5 present respectively applications of MsFEMs in two-dimensional non-periodic or periodic heterogeneous media. Section 6.6 presents some applications of MsFEMs in three-dimensional media.

6.1 Notations

We recall that in the Crouzeix-Raviart MsFEM defined by Stokes equations (see section 3.2) or Oseen equations (see section 3.3), weighting functions are defined by (3.7) and we choose in this chapter: for any $E \in \mathcal{E}_H$,

$$\text{for } d=2: \begin{cases} s=2, \ \boldsymbol{\omega}_{E,1}=\boldsymbol{e}_1, \ \boldsymbol{\omega}_{E,2}=\boldsymbol{e}_2. \\ s=3, \ \boldsymbol{\omega}_{E,1}=\boldsymbol{e}_1, \ \boldsymbol{\omega}_{E,2}=\boldsymbol{e}_2, \ \boldsymbol{\omega}_{E,3}=\boldsymbol{n}_E\psi_E. \\ s=4, \ \boldsymbol{\omega}_{E,1}=\boldsymbol{e}_1, \ \boldsymbol{\omega}_{E,2}=\boldsymbol{e}_2, \ \boldsymbol{\omega}_{E,3}=\boldsymbol{n}_E\psi_E, \ \boldsymbol{\omega}_{E,4}=\boldsymbol{\tau}_E\phi_E. \end{cases} \tag{6.1}$$

$$\text{for } d=3: \ \boldsymbol{\omega}_{E,1}=\boldsymbol{e}_1, \ \boldsymbol{\omega}_{E,2}=\boldsymbol{e}_2, \ \boldsymbol{\omega}_{E,3}=\boldsymbol{e}_3, \ \boldsymbol{\omega}_{E,4}=\boldsymbol{n}_E\varphi_E. \tag{6.2}$$

where $\{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_d\}$ is the canonical basis of $\mathbb{R}^d$ and $\boldsymbol{n}_E$, $\boldsymbol{\tau}_E$ are respectively the unit vector normal and tangent to the face E. Besides, ψ_E, ϕ_E and φ_E are linear polynomials which satisfy that $\int_{E\cap\Omega^\varepsilon}\psi_E=0$, $\int_{E\cap\Omega^\varepsilon}\phi_E=0$ and $\int_{E\cap\Omega^\varepsilon}\varphi_E=0$. In numerical simulations, we choose

$$\text{for } d=2: \begin{cases} \psi_E = 2x+4y+c_1 \\ \phi_E = 5x-3y+c_2 \end{cases} \tag{6.3}$$

$$\text{for } d=3: \ \varphi_E = 4x+4y-8z+c_3 \tag{6.4}$$

where c_1, c_2 are computed respectively by the relation $\int_{E\cap\Omega^\varepsilon}\psi_E=0$ and $\int_{E\cap\Omega^\varepsilon}\phi_E=0$. For $d=3$, c_3 is computed by the relation $\int_{E\cap\Omega^\varepsilon}\varphi_E=0$.

Throughout this chapter, the term CR2, CR3, CR4 denote respectively Crouzeix-Raviart MsFEM defined by Stokes equations with weighting functions in the case of $s=2$, $s=3$ and $s=4$ defined by (6.1). For $d=3$, weighting functions are chosen as (6.2) with the polynomial defined by (6.4). We still denote this multiscale method by CR3 in three dimensions. Besides, the term CR2_Stab, CR3_Stab represent respectively the CR2, CR3 stabilized by the SUPG technique presented in subsection 3.2.4. The stabilized versions are used to solve Navier-Stokes problems on the coarse mesh.

Similarly, the term CR2_Os, CR3_Os, CR4_Os stand for respectively Crouzeix-Raviart MsFEM defined by Oseen equations with weighting functions in the case of $s=2$, $s=3$ and $s=4$ defined by (6.1).

We use {CR2+CR2_Os} and {CR2+CR2_Os}_Stab to stand for respectively the Crouzeix-Raviart MsFEM defined by adding solutions of both local Stokes problems and local Oseen problems (see section 3.4), and its stabilized version by the SUPG technique. Weighting functions are chosen as in the case of $s=2$ defined by (6.1).

The term {CR2+B}_Stab stands for the stabilized Crouzeix-Raviart MsFEM defined by Stokes equations and enriched by bubble functions (see section 3.5).

We recall that in the high-order Crouzeix-Raviart MsFEM defined by Stokes equations (see section 3.6) or Oseen equations (see section 3.7), weighting functions in two dimensions are defined by (3.203) and we choose in this chapter: for any $T \in \mathcal{T}_H$ and for any $E \in \mathcal{E}(T)$,

$$\text{for } d=2: \begin{cases} s=4: \ \boldsymbol{\omega}_{E,1}=\boldsymbol{e}_1, \boldsymbol{\omega}_{E,2}=\boldsymbol{e}_2, \ \boldsymbol{\omega}_{E,3}=\boldsymbol{n}_E\psi_E, \ \boldsymbol{\omega}_{E,4}=\boldsymbol{\tau}_E\phi_E. \\ t=3: \ \varpi_{T,1}=1, \ \varpi_{T,2}=x, \ \varpi_{T,3}=y. \\ r=2: \ \boldsymbol{\varphi}_{T,1}=\boldsymbol{e}_1, \boldsymbol{\varphi}_{T,2}=\boldsymbol{e}_2. \end{cases} \tag{6.5}$$

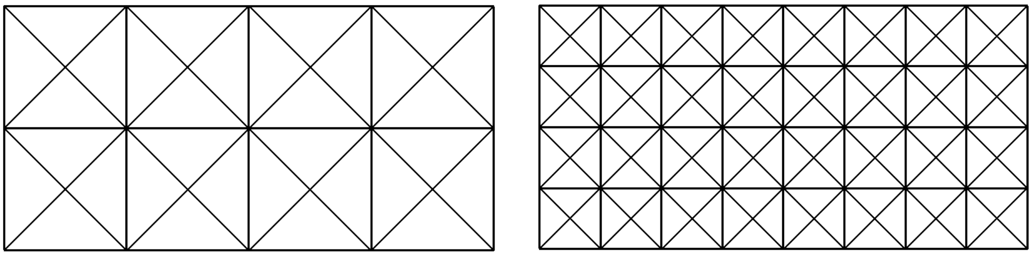
where $\{\boldsymbol{e}_1, \boldsymbol{e}_2\}$ is the canonical basis of $\mathbb{R}^2$ and $\boldsymbol{n}_E$, $\boldsymbol{\tau}_E$ are respectively the unit vector normal and tangent to the face E. Besides, ψ_E and ϕ_E are linear polynomials which satisfy that $\int_{E\cap\Omega^\varepsilon}\psi_E=0$ and $\int_{E\cap\Omega^\varepsilon}\phi_E=0$. In practice, ψ_E and ϕ_E are chosen the same as in (6.3).

We use the term CR4_high to denote the high-order Crouzeix-Raviart MsFEM defined by

Stokes equations with weighting functions defined by (6.5). Note that we have discretized local problems with the Crouzeix-Raviart finite element presented in subsection 4.1.1 or the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element presented in subsection 4.6.1. We use respectively CR4_high_P0 and CR4_high to denote the high-order MsFEM where local problems are discretized by the Crouzeix-Raviart finite element or the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element. The term CR4_Os_high stands for the high-order Crouzeix-Raviart MsFEM based on Oseen equations where local problems are discretized by the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element. Relative errors of the velocity and pressure are computed using formulas defined by (5.1).

6.2 Simulations in a two-dimensional homogeneous medium

We solve Stokes flows in a two-dimensional homogeneous channel $\Omega = [0, 2] \times [0, 1]$. We assign $\rho = 1$, $\mu = 1$ and the source term $\boldsymbol{f} = \mathbf{0}$. A parabolic velocity profile $\boldsymbol{u} = y\,(1 - y)\,\boldsymbol{e}_1$ is imposed at the inlet. The boundary condition $\nabla \boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \mathbf{0}$ is imposed at the outlet. The no-slip boundary condition is imposed on other boundaries.



Config. 2×4 Config. 4×8

FIGURE 6.1: Examples of coarse mesh

TABLE 6.1: Error analysis of Stokes flows in a homogeneous medium

Config.	L^2 rel. U				L^2 rel. P			
	CR2	CR3	CR4	CR4_high	CR2	CR3	CR4	CR4_high
2×4	0.1573	0.0139	6.3e-4	3.e-5	0.1875	0.0169	0.0047	8.1e-4
4×8	0.0648	0.0036	5.7e-4	1.4-5	0.0485	0.0072	0.0019	4.6e-4
8×16	0.0131	9.4e-4	1.2e-4	1.0e-5	0.0232	0.0035	6.7e-4	3.3e-4
16×32	0.0034	7.0e-4	9.2e-5	9.2e-6	0.0097	0.0015	5.4e-4	2.5e-4
32×64	8.2e-4	6.2e-4	8.7e-5	8.8e-6	0.0048	0.0010	4.6e-4	2.0e-4

In Table 6.1, the first column Config. indicates the partition of the domain Ω, i.e. the coarse mesh $\mathcal{T}_H$ as shown in Figure 6.1. Table 6.1 reveals relative errors of velocity and pressure on a number of coarse meshes, showing a convincingly converging trends with respect to H. It is highlighted that CR4_high is the most accurate in both the velocity and pressure. Figure 6.2 shows that the velocity and pressure given by CR2 converges with rate about 2 and 1 respectively. This is coherent with the convergence rate of the conventional Crouzeix-Raviart finite element. The errors of other methods are rather small even on the coarse mesh 2×4. Figure 6.3 and Figure 6.4 present respectively the velocity and pressure contour computed on the coarse mesh 2×4 shown in Figure 6.1. We observe that the velocity and pressure computed by CR4 and CR4_high are already very close to the reference solutions even on the coarse mesh 2×4. However, the velocity and pressure computed by CR2 are much less accurate.

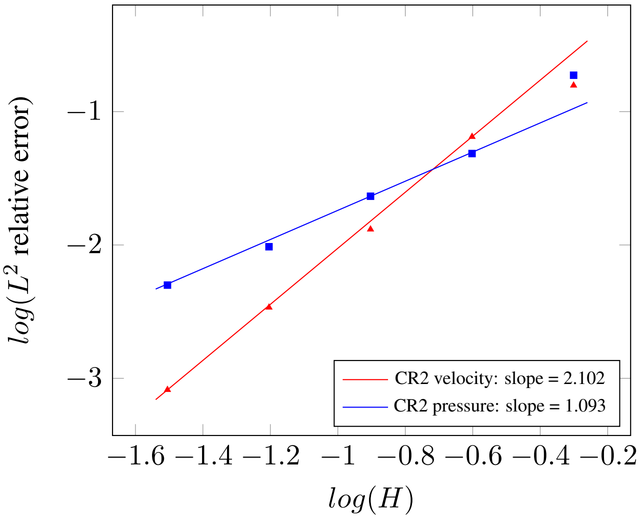
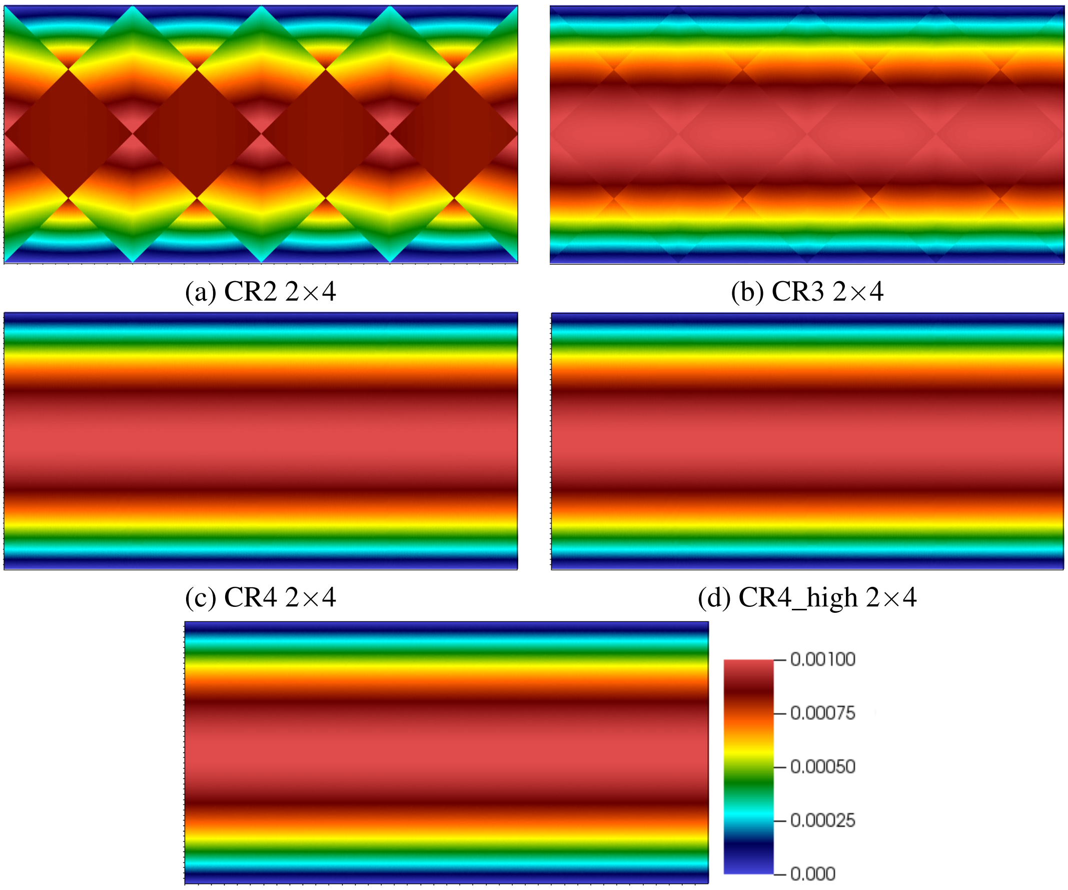


FIGURE 6.2: Numerical convergence of CR2 in a homogeneous medium


(a) CR2 2×4

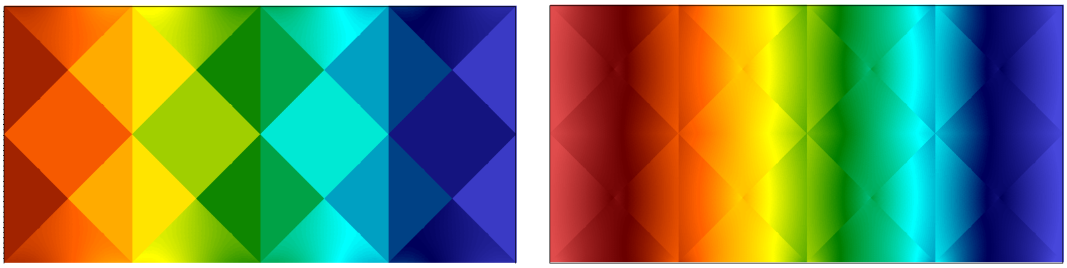
(b) CR3 2×4

(c) CR4 2×4

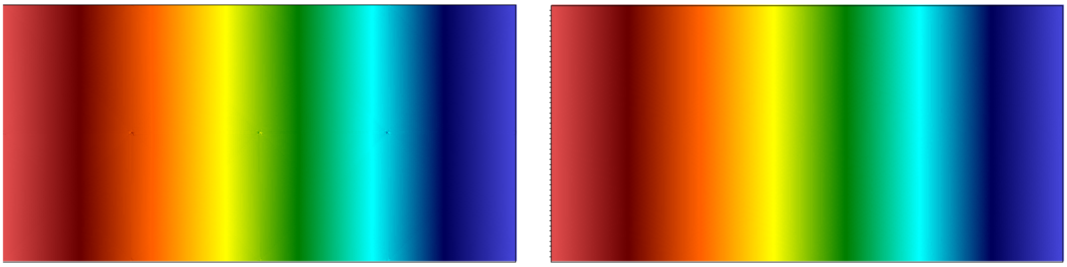
(d) CR4_high 2×4

(e) reference

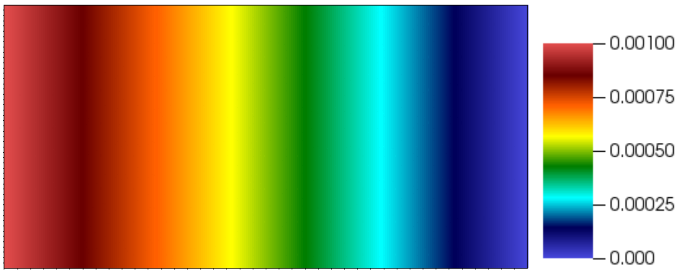
FIGURE 6.3: $|\boldsymbol{u}|$ contours of Stokes flows in the homogeneous medium



(a) CR2 2×4 (b) CR3 2×4

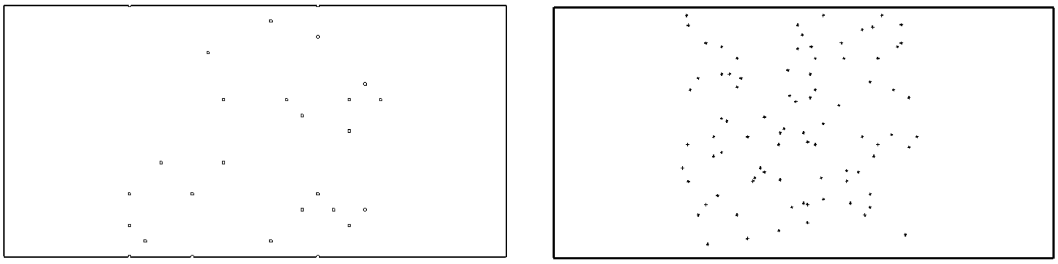


(c) CR4 2×4 (d) CR4_high 2×4



(e) reference

FIGURE 6.4: Pressure contours of Stokes flows in the homogeneous medium



case (A) case (B)

FIGURE 6.5: A description of computational domains

6.3 Numerical convergence in the periodic case

In practical implementation of multiscale finite element methods, we do not make any assumption on the type of heterogeneities of the media. However, it is more convenient to analyze the convergence of the multiscale methods in the context of periodically placed solid obstacles. The essential reason for this restriction is that an accurate description of the asymptotic behavior (as $\varepsilon \to 0$) of the exact solution $\boldsymbol{u}^\varepsilon$ is needed in the numerical analysis. Such an accurate description is available in the periodic setting. It is provided by the two-scale asymptotic expansion (see section 1.3) of the homogenized solution to the problem. Now we present some convergence results of the literature which are obtained in the periodic setting.

Let $(\boldsymbol{u}^\varepsilon, p^\varepsilon) \in \left(H_0^1(\Omega^\varepsilon)\right)^d \times L_0^2(\Omega^\varepsilon)$ be the sufficiently smooth exact solution to Stokes problem (1.2) and let $(\boldsymbol{u}_H, p_H) \in X_H$ be the discrete solution computed by multiscale finite element methods. In the work of [95], the authors suppose that $\boldsymbol{f}$ and the homogenized pressure p^* are sufficiently smooth in a domain with periodically placed perforations. It is shown that the upper error bound between exact solution and its MsFEM approximation is

$$\begin{aligned} &\|\boldsymbol{u}^\varepsilon - \boldsymbol{u}_H\|_{H^1(\Omega)} + \varepsilon \|p^\varepsilon - p_H\|_{L^2(\Omega)} \\ &\qquad \leq C\varepsilon \left(H + \sqrt{\varepsilon} + \sqrt{\frac{\varepsilon}{H}}\right) \left(\|\boldsymbol{f}\|_{H^2(\Omega)\cap C^1(\bar{\Omega})} + \|p^*\|_{H^2(\Omega)}\right) \end{aligned} \tag{6.6}$$

where the constant C depends only on the mesh regularity and the perforation pattern. The relative error of velocity is thus at the order of $\left(H + \sqrt{\varepsilon} + \sqrt{\frac{\varepsilon}{H}}\right)$. This error bound does not take into account the error related to the discretization of local problems on fine meshes by supposing that $h \ll \varepsilon$.

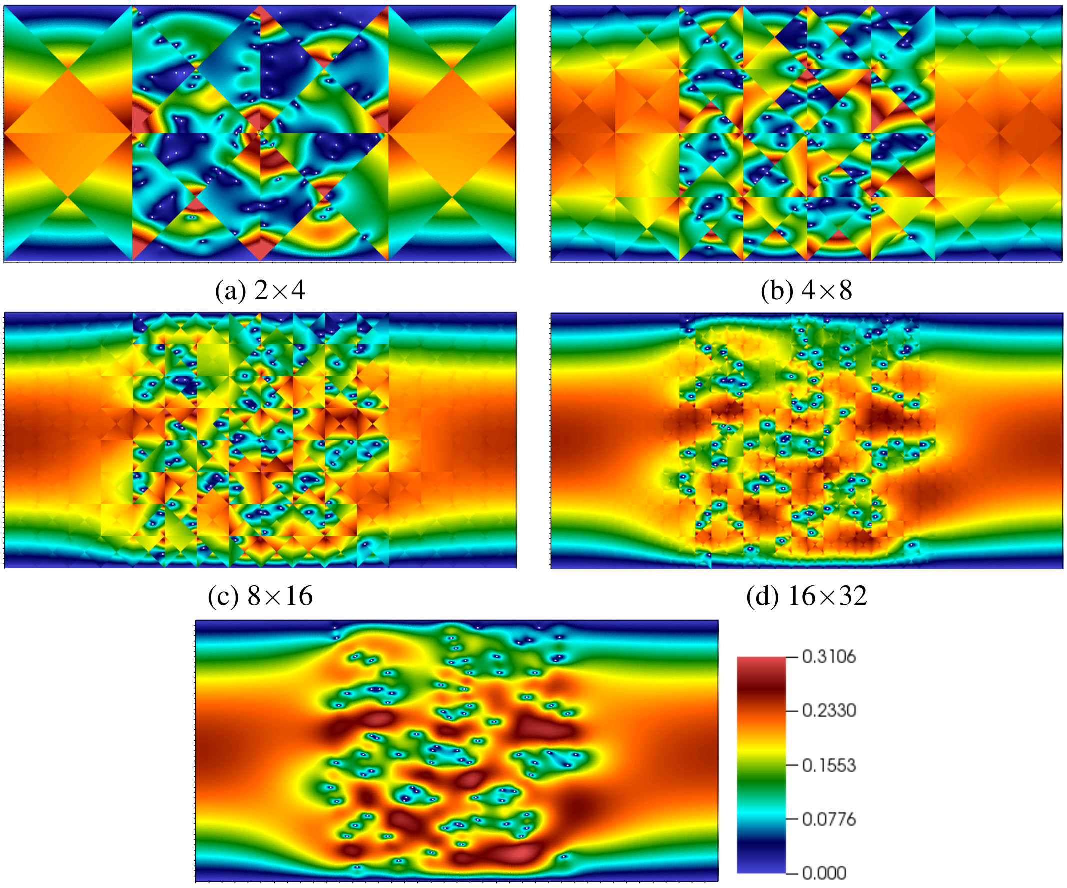
In numerical convergence studies presented in this chapter, we fix the element size h of fine meshes, the size of obstacles ε and vary only the element size H of the coarse mesh. We ensure that these three sizes satisfy $h \ll \varepsilon < H$. The element size h verifies $h \ll \varepsilon$ in order to capture precisely small obstacles in the media. We are interested only in the case $H > \varepsilon$ since the opposite case $H < \varepsilon$ is covered by classical finite element methods and the multiscale finite element method is not needed. It is easy to see that when H decreases, $\sqrt{\frac{\varepsilon}{H}}$ increases. Thus we can not expect in numerical experiments that the velocity error converges to 0 in the limit when $H \to 0$.

6.4 Simulations in two-dimensional non-periodic heterogeneous media

In this section, we consider a channel domain $\Omega = [0, 2] \times [0, 1]$ with two cases of arbitrarily placed obstacles. As shown in Figure 6.5, the case (A) consists of about 26 obstacles of diameter $\varepsilon = 0.015$ and the case (B) consists of about 100 obstacles of diameter $\varepsilon = 0.003$. We present applications of different Crouzeix-Raviart MsFEMs to solve Stokes problems, Oseen problems and Navier-Stokes problems in both cases (A) and (B).

The reference solutions in the case (A) of Figure 6.5 are computed on a reference mesh consisting of about 600,000 fine elements with an average element size $h = 0.002$. Using the *prediction-correction* algorithm in *TrioCFD*, the number of degrees of freedom are respectively about 1,800,000 and 600,000 in the prediction step and the correction step. Note that the number of faces is about 1.5 times the number of elements in the reference mesh. In the prediction step, the degrees of freedom are velocities located on faces of the mesh and thus the number of degrees of freedom is the product of number of faces and the number of components of the velocity. The reference solutions in the case (B) are solved on a reference mesh consisting of about 2,000,000 elements with an average element size $h = 0.00025$. The reference velocity contour in the case

(B) is shown in Figure 6.6. In the prediction step and the correction step, the number of degrees of freedom are respectively about 6,000,000 and 2,000,000.



(a) 2×4

(b) 4×8

(c) 8×16

(d) 16×32

(e) reference

FIGURE 6.6: $|\boldsymbol{u}|$ contours of Stokes flows in the case (B) of Figure 6.5 computed by CR2

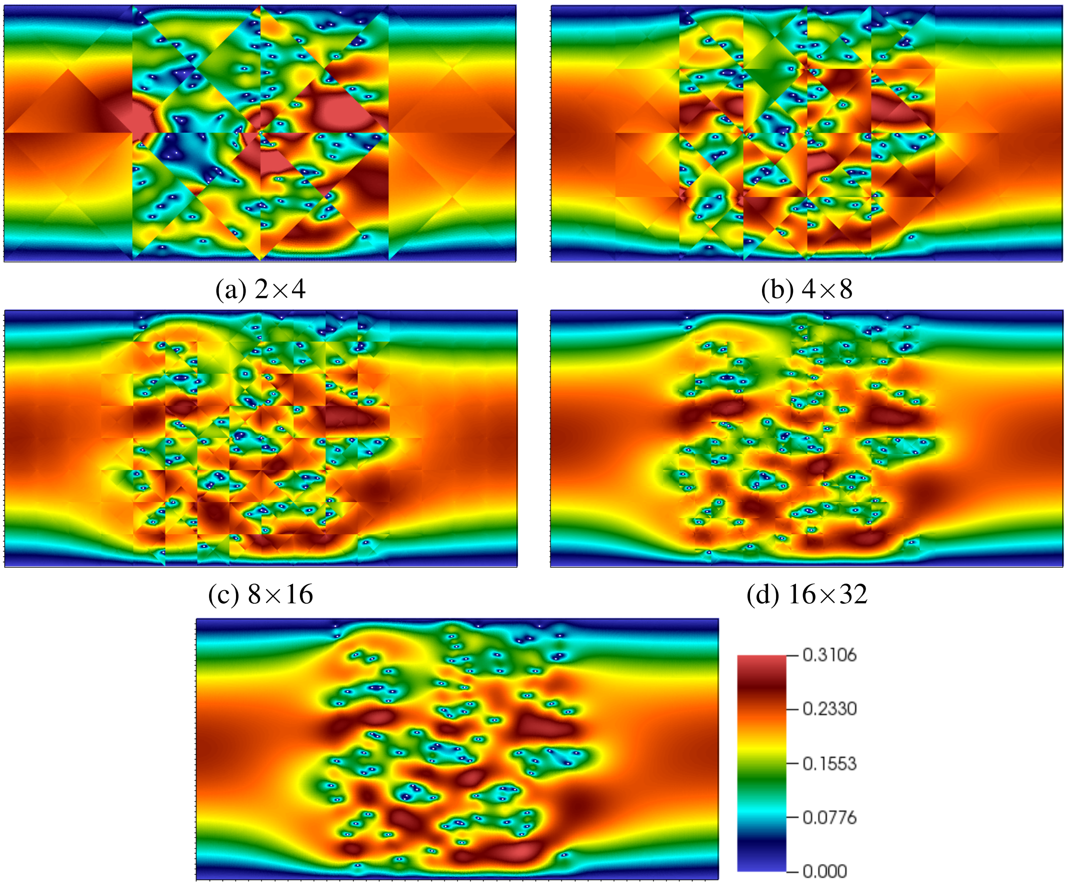
6.4.1 Applications to Stokes flows

We first apply some Crouzeix-Raviart MsFEMs to solve Stokes problems on the coarse mesh. The nonlinear convection term of Navier-Stokes problems is thus neglected. The simplicity of Stokes problems allows to perform numerical tests easily and to compare the accuracy and performance of different MsFEMs.

6.4.1.1 Numerical convergence of Crouzeix-Raviart MsFEMs with respect to H

We assign $\rho = 1$, $\mu = 1$ and the source term $\boldsymbol{f} = 0$. A parabolic velocity $\boldsymbol{u} = y\,(1-y)\,\boldsymbol{e}_1$ is imposed at the inlet. The boundary condition $\nabla \boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{0}$ is imposed at the outlet. The no-slip boundary condition is imposed on other boundaries. In this study, we fix the element size h of fine-scale meshes while decreasing the element size H of the coarse-scale mesh.

Figure 6.6, Figure 6.7 and Figure 6.8 reveal the reconstructed fine-scale velocity computed respectively by CR2, CR3 and CR4_high on several coarse meshes. The velocity computed by CR4_high and CR4_high_P0 are the same and thus only CR4_high is presented here. Comparing velocity fields shown in Figures 6.6 to 6.8, we conclude that the velocity computed by Crouzeix-Raviart multiscale methods converges well to the reference solution. The presence of very small obstacles and important flow features in the heterogeneous medium are already well captured on



(a) 2×4 (b) 4×8

(c) 8×16 (d) 16×32

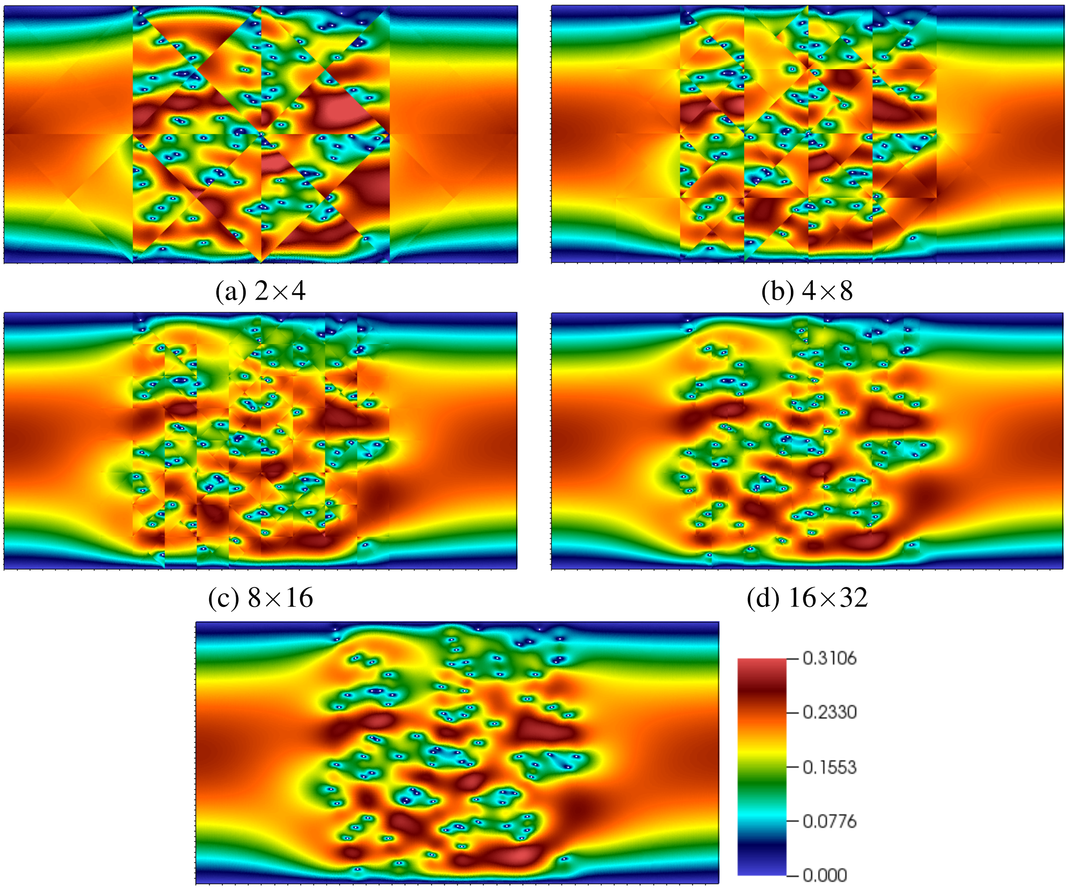
(e) reference

FIGURE 6.7: $|\boldsymbol{u}|$ contours of Stokes flows in the case (B) of Figure 6.5 computed by CR3

the coarse mesh 8×16 or 16×32, at a rather large H/ε ratio. Figure 6.9 compares the velocity contour computed by CR3 and CR4 on the coarse mesh 8×16. We see that CR4 is slightly more accurate than CR3.

On combining relative velocity errors shown in Table 6.2 and Table 6.3, we find that (i) multiscale methods which use more weighting functions in the approximation space of velocity, such as CR3, CR4 and CR4_high are significantly more accurate than CR2; (ii) CR4_high is more accurate than both CR3 and CR4 and this confirms the superiority of high-order Crouzeix-Raviart multiscale methods. Figure 6.10 and Figure 6.11 show the numerical convergence of velocity computed by different MsFEMs in both case (A) and (B). It is shown that CR2, CR3, CR4 and CR4_high have almost the same rate of convergence, which is about 1. However, CR4_Os_high has the smallest errors in both case (A) and (B). The same phenomenon has also been observed in the high-order method proposed in [50]. In this thesis, only CR4 high with $n = 1$ has been implemented and tested. In the future work, an error analysis could be performed for the high-order method in order to find the relation between error of the method and the degree of polynomials n. It would be also interesting to implement the high-order MsFEMs for $n = 2$, $n = 3$ and so on and compare its performance with other Crouzeix-Raviart MsFEMs.

Figures 6.12 to 6.14 show the reconstructed fine-scale pressure computed respectively by CR2, CR3 and CR4_high on several coarse meshes. It is revealed that the pressure computed by Crouzeix-Raviart multiscale methods converges well toward the reference pressure. Most important fine-scale features and the presence of small obstacles are already well captured on coarse meshes with a rather large H/ε ratio. Globally, the pressure computed by CR4_high is

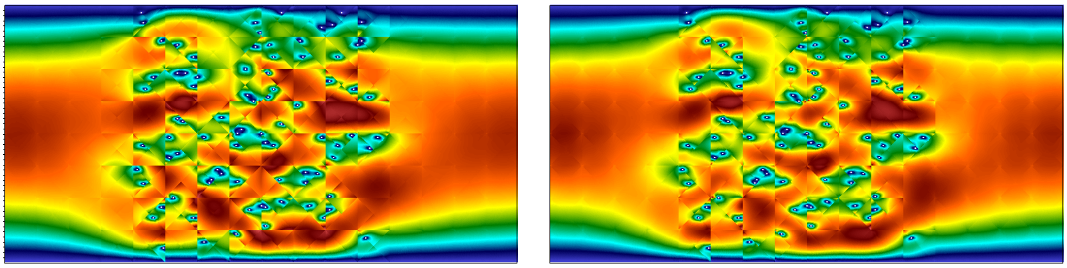


(a) 2×4 (b) 4×8

(c) 8×16 (d) 16×32

(e) reference

FIGURE 6.8: $|\boldsymbol{u}|$ contours of Stokes flows in the case (B) of Figure 6.5 computed by CR4_high



(c) CR3 8×16 (d) CR4 8×16

FIGURE 6.9: $|\boldsymbol{u}|$ contours of Stokes flows in the case (B) of Figure 6.5 computed by CR3 and CR4

TABLE 6.2: Error analysis of Stokes flows in the case (A) of Figure 6.5

Config.	H/ε	L^2 Rel. U				L^2 Rel. P			
		CR2	CR3	CR4	CR4_high	CR2	CR3	CR4	CR4_high
2×4	50	0.334	0.210	0.190	0.091	0.475	0.212	0.146	0.096
4×8	25	0.174	0.072	0.065	0.039	0.321	0.144	0.083	0.055
8×16	12.5	0.061	0.026	0.020	0.015	0.173	0.075	0.044	0.034
16×32	6.25	0.030	0.014	0.012	0.006	0.116	0.053	0.025	0.020

TABLE 6.3: Error analysis of Stokes flows in the case (B) of Figure 6.5

Config.	H/ε	L^2 Rel. U				L^2 Rel. P			
		CR2	CR3	CR4	CR4_high	CR2	CR3	CR4	CR4_high
2×4	125	0.478	0.301	0.274	0.159	0.595	0.315	0.281	0.227
4×8	62.5	0.329	0.185	0.158	0.114	0.556	0.289	0.249	0.189
8×16	31.25	0.189	0.096	0.084	0.061	0.449	0.232	0.164	0.130
16×32	15.63	0.116	0.055	0.045	0.030	0.307	0.184	0.118	0.102

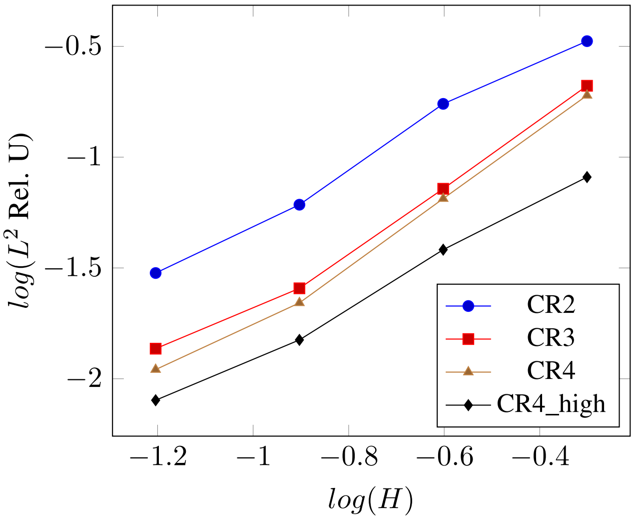


FIGURE 6.10: Numerical convergence of MsFEMs in the case (A) of Figure 6.5

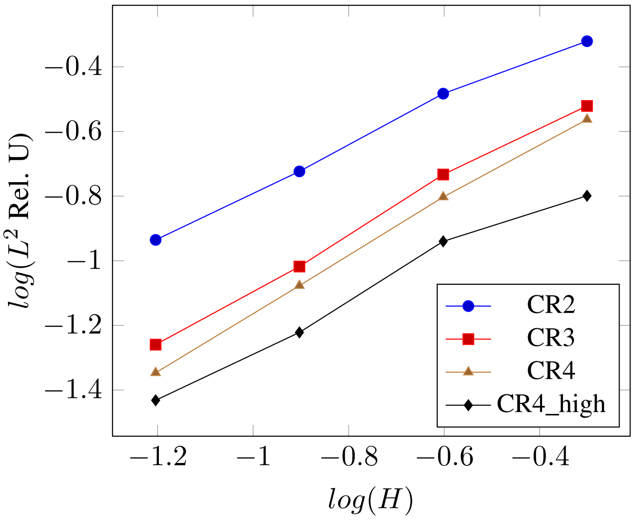
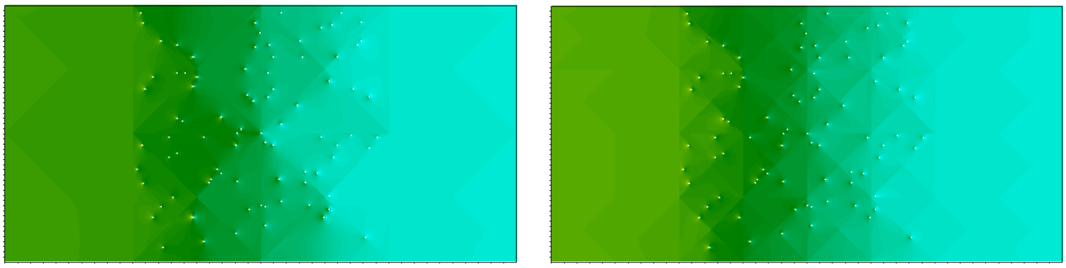
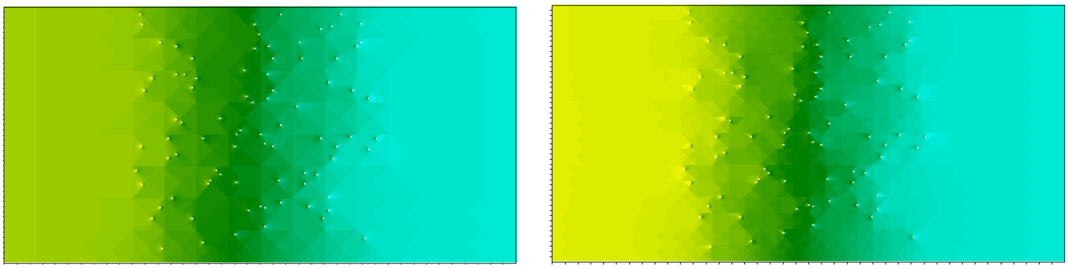


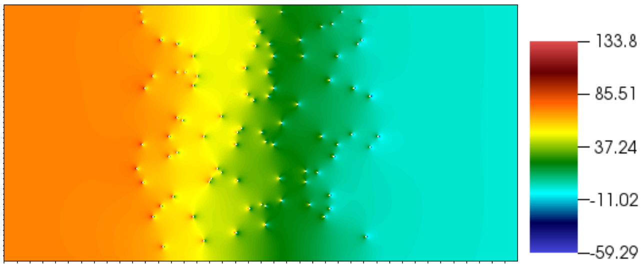
FIGURE 6.11: Numerical convergence of MsFEMs in the case (B) of Figure 6.5



(a) 2×4 (b) 4×8

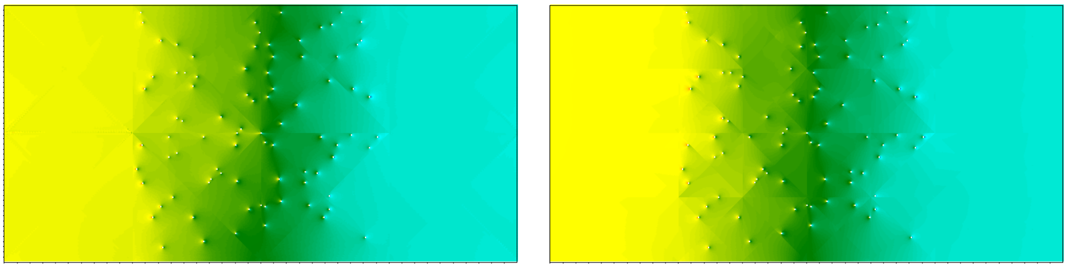


(c) 8×16 (d) 16×32

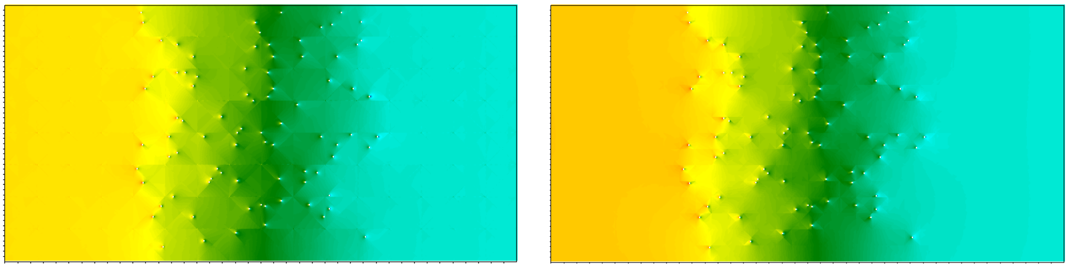


(e) reference

FIGURE 6.12: Pressure contours of Stokes flows in the case (B) of Figure 6.5 computed by CR2

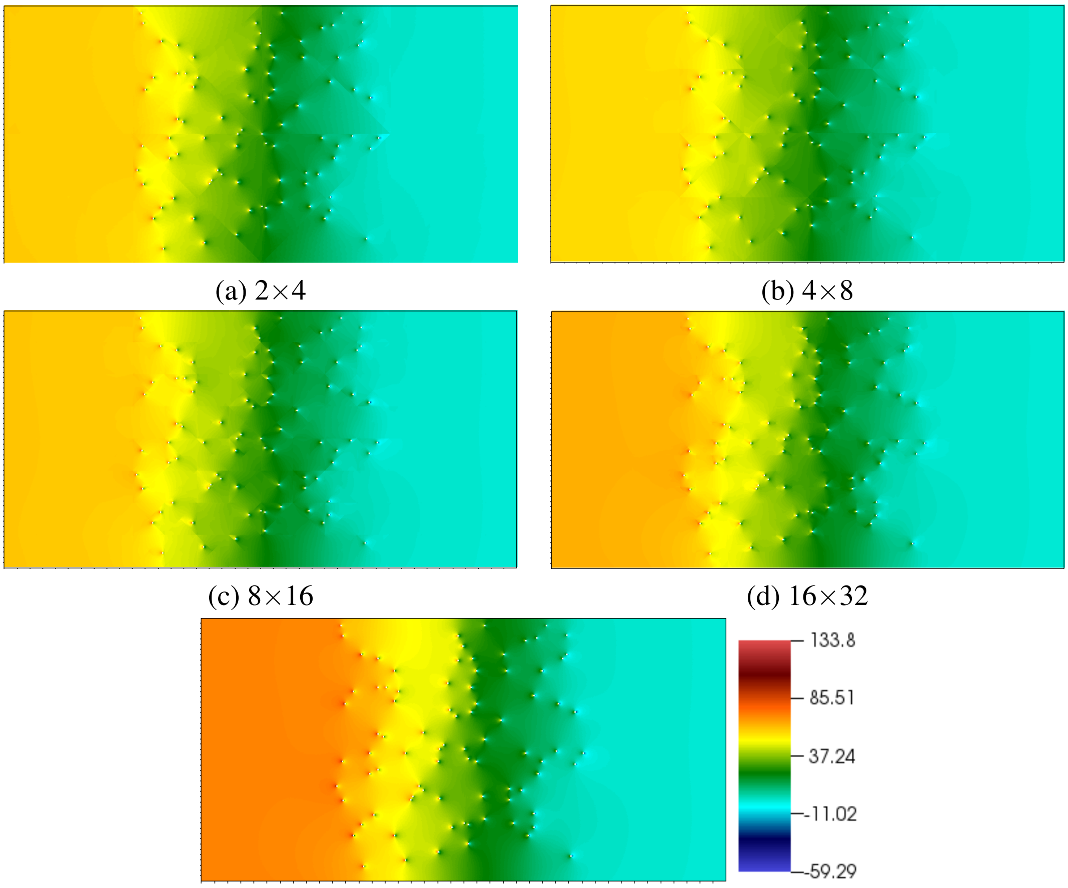


(a) 2×4 (b) 4×8



(c) 8×16 (d) 16×32

FIGURE 6.13: Pressure contours of Stokes flows in the case (B) of Figure 6.5 computed by CR3

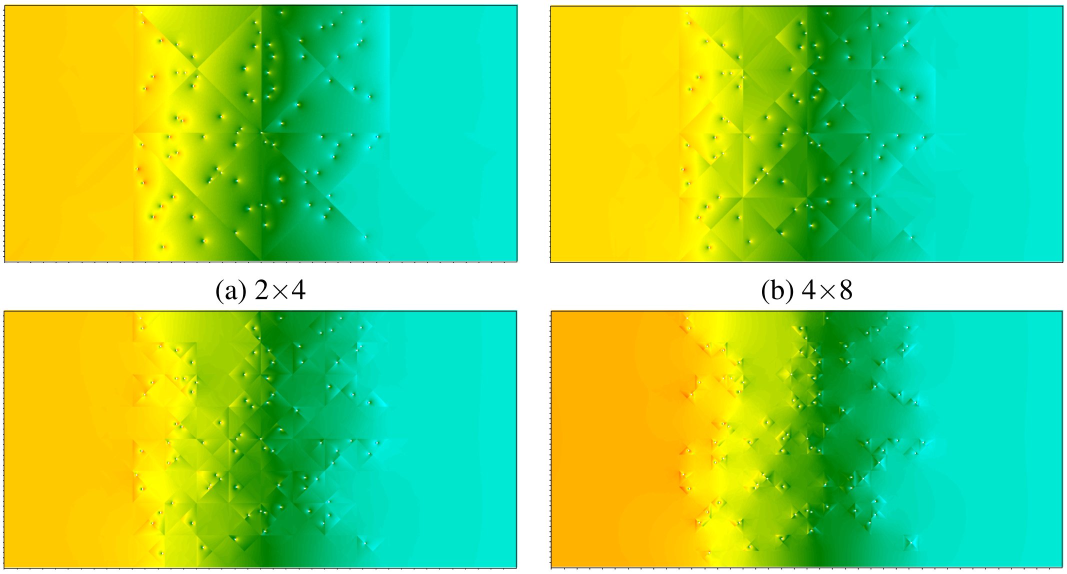


(a) 2×4 (b) 4×8

(c) 8×16 (d) 16×32

(e) reference

FIGURE 6.14: Pressure contours of Stokes flows in the case (B) of Figure 6.5 computed by CR4_high



(a) 2×4 (b) 4×8

(c) 8×16 (d) 16×32

FIGURE 6.15: Pressure contours of Stokes flows in the case (B) of Figure 6.5 computed by CR4_high_P0

evidently more accurate compared to that of CR2 and CR3 shown in Figure 6.13. Remarkably, as shown in Figure 6.14, the pressure computed by CR4_high seems already rather satisfying on the coarse mesh 2×4.

Figure 6.15 shows the pressure computed by CR4_high_P0 on several coarse meshes. Comparing Figure 6.15 and Figure 6.14, we conclude that discretizing the pressure of local problems in the $\mathbb{P}_1$ space instead of $\mathbb{P}_0$ improves the accuracy of the pressure solution. This result underlines the importance of the type of finite element used to discretize local problems in Crouzeix-Raviart MsFEMs.

6.4.1.2 The Crouzeix-Raviart MsFEM enriched by bubble functions

In the case (B) of Figure 6.5, we impose $\boldsymbol{u} = \mathbf{0}$ on $\partial\Omega$ and the source term $\boldsymbol{f} = (1/2 - y, x - 1/2)$. We solve a Stokes problem in the case (B) of Figure 6.5 on several coarse meshes using CR2_Stab and {CR2+B}_Stab. In practice, it is necessary to use the SUPG method to preclude oscillations in the solution.

TABLE 6.4: Error analysis of the MsFEM enriched by bubble functions in the case (B) of Figure 6.5

Config.	H/ε	L^2 Rel. U		L^2 Rel. P	
		CR2_Stab	{CR2+B}_Stab	CR2_Stab	{CR2+B}_Stab
2×4	125	0.478	0.481	0.595	0.602
4×8	62.5	0.329	0.334	0.556	0.563
8×16	31.25	0.189	0.196	0.449	0.459
16×32	15.63	0.143	0.149	0.307	0.316

Contrary to expectations, Table 6.4 shows that the addition of bubble functions defined in section 3.5 cannot improve the accuracy of CR2. This agrees well with our mathematical difficulties in the analysis of the method in subsection 3.5.2. Compared to the accuracy of CR4_high, this numerical result confirms that CR4_high provides the correct definition of bubble functions for Stokes equations which can effectively improve the accuracy of Crouzeix-Raviart multiscale methods.

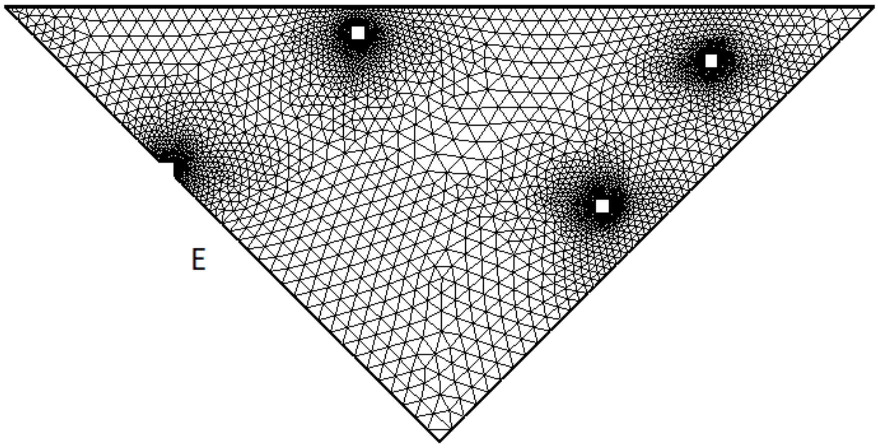
6.4.1.3 Multiscale basis functions defined by Stokes equations

Figure 6.16 (A) shows a coarse element generated in the case (B) of Figure 6.5. Figure 6.16 (B) and (C) and Figure 6.17 show the multiscale basis functions associated to the boundary E computed respectively by CR2 and CR3. We recall that without obstacles, multiscale basis functions computed by CR2 are actually the Crouzeix-Raviart finite element basis functions (see subsection 4.1.1). However, with the presence of obstacles, Figures 6.16 to 6.17 show that multiscale basis functions computed with CR2 and CR3 are completely different from polynomial basis functions used in classical finite element methods.

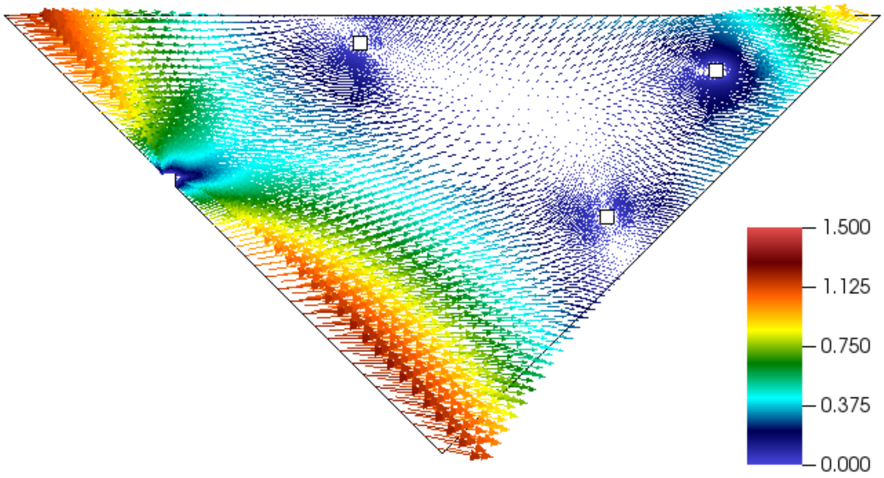
We observe that multiscale basis functions capture very well the presence of obstacles on the coarse element and the no-slip boundary condition is well imposed on the boundary of obstacles. Even on the boundary E which is cut by obstacles, the integral condition $\int_E \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{\omega}_{E,j} = \delta_{ij}$ for $i, j = 1, \cdots, 2$ is successfully satisfied.

6.4.1.4 Computing time of different steps of multiscale methods

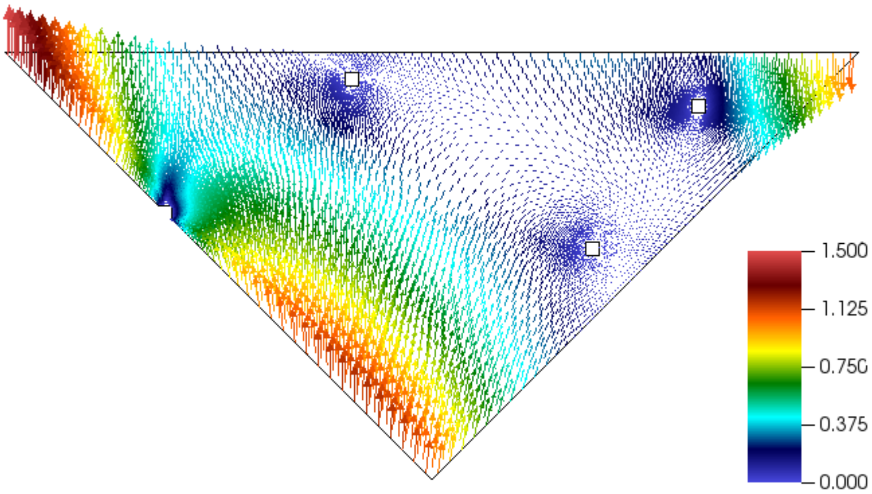
In this thesis, numerical simulations are performed on the cluster *Callisto* at CEA Saclay. We have used mainly two kinds of computing node: *Eris* and *Pluton*. There are in total 36 nodes



(A) The coarse element with the boundary E

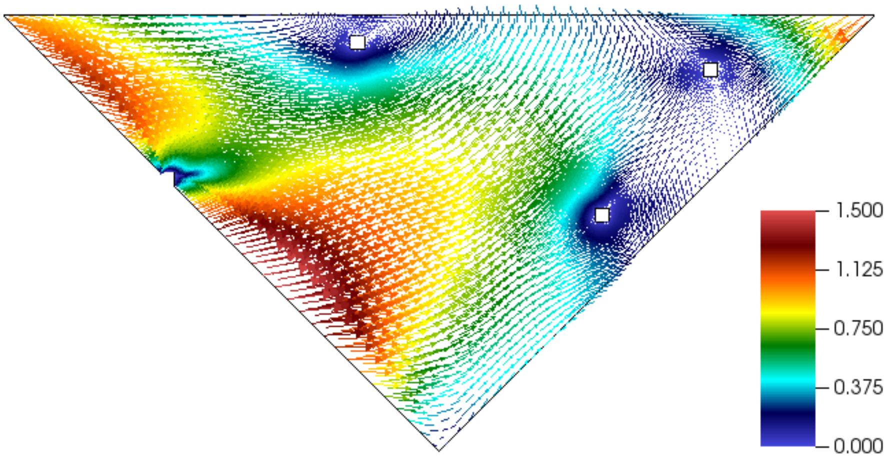


(B) Vector $\mathbf{\Phi}_{E,1}$

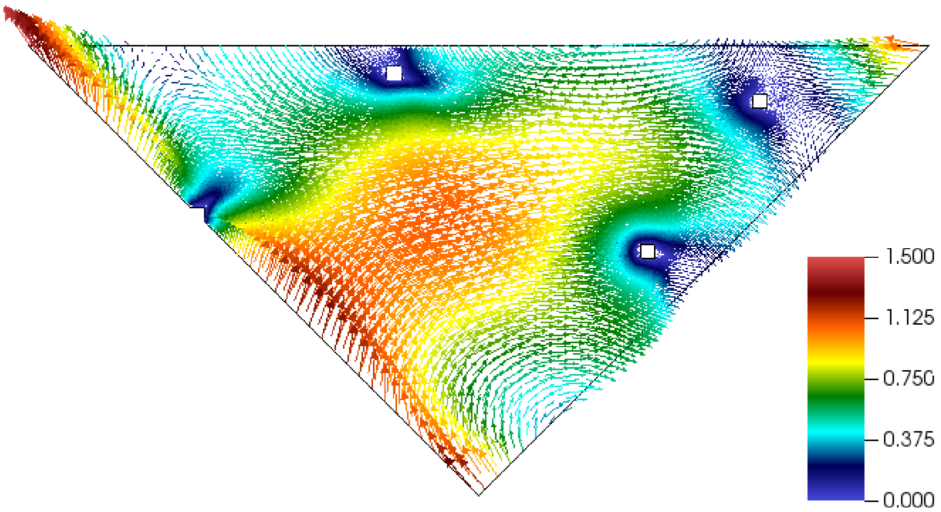


(C) Vector $\mathbf{\Phi}_{E,2}$

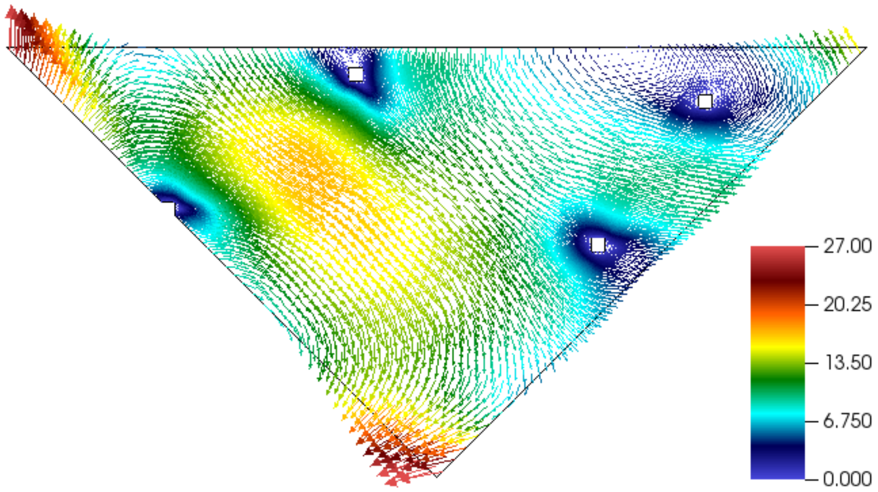
FIGURE 6.16: The coarse element and multiscale basis functions computed by CR2



(A) Vector $\mathbf{\Phi}_{E,1}$



(B) Vector $\mathbf{\Phi}_{E,2}$



(C) Vector $\mathbf{\Phi}_{E,3}$

FIGURE 6.17: Multiscale basis functions computed by CR3

of *Eris*, each with two CPUs of type Intel(R) Xeon(R) X5667 (frequency 3.06 GHz, memory 48 Go per CPU) and each CPU consists of four cores. We have at our disposal at most a total of 288 cores. There are in total 28 nodes of *Pluton*, each with two CPUs of type Intel(R) Xeon(R) L5640 (frequency 2.26 GHz, memory 36 Go per CPU) and each CPU consists of 6 cores. We have at our disposal at most a total of 336 cores. Overall, we have at our disposal at most 624 cores.

Now we present the computing time used by CR3 for solving some local problems on the coarse mesh 8×16 in the case (B) of Figure 6.5. There are in average about 5000 small elements in each fine mesh. Using the *prediction-correction* algorithm, the number of degrees of freedom in the prediction step and the correction step are respectively about 15,000 and 5000. The computing time for solving one local problem sequentially is between $900\,\mathrm{s}$ and $1200\,\mathrm{s}$. Some local problems take more computing time due to the complexity of coarse elements or due to the small memory of some computing nodes in the cluster. At the end of this thesis, we implemented a direct solver to solve same local problems in only several seconds. This reduces significantly the time for solving local problems and improves the performance of multiscale finite element methods.

The average time used by CR3 for computing matrices and reconstructing fine-scale solutions on one coarse element is about $0.03\,\mathrm{s}$ which is negligible. The solution of coarse-scale problems with the *prediction-correction* algorithm takes maximum $300\,\mathrm{s}$ on the coarse mesh 8×16.

Making use of all available processors on *Callisto*, the total time for solving local problems with the *prediction-correction* algorithm is about $120\,\mathrm{minutes}$. The total time for computing matrices and fine-scale solutions on all coarse elements is about $10\,\mathrm{minutes}$. It takes about $12\,\mathrm{hours}$ to compute reference solutions in parallel using 80 processors. We recall that reference solutions are computed by using the finite volume element method and the *prediction-correction* algorithm in *TrioCFD*. As a result, for solving the same flow problem in both cases (A) and (B) of Figure 6.5, classical numerical methods take up to 6 times more computing time than MsFEMs.

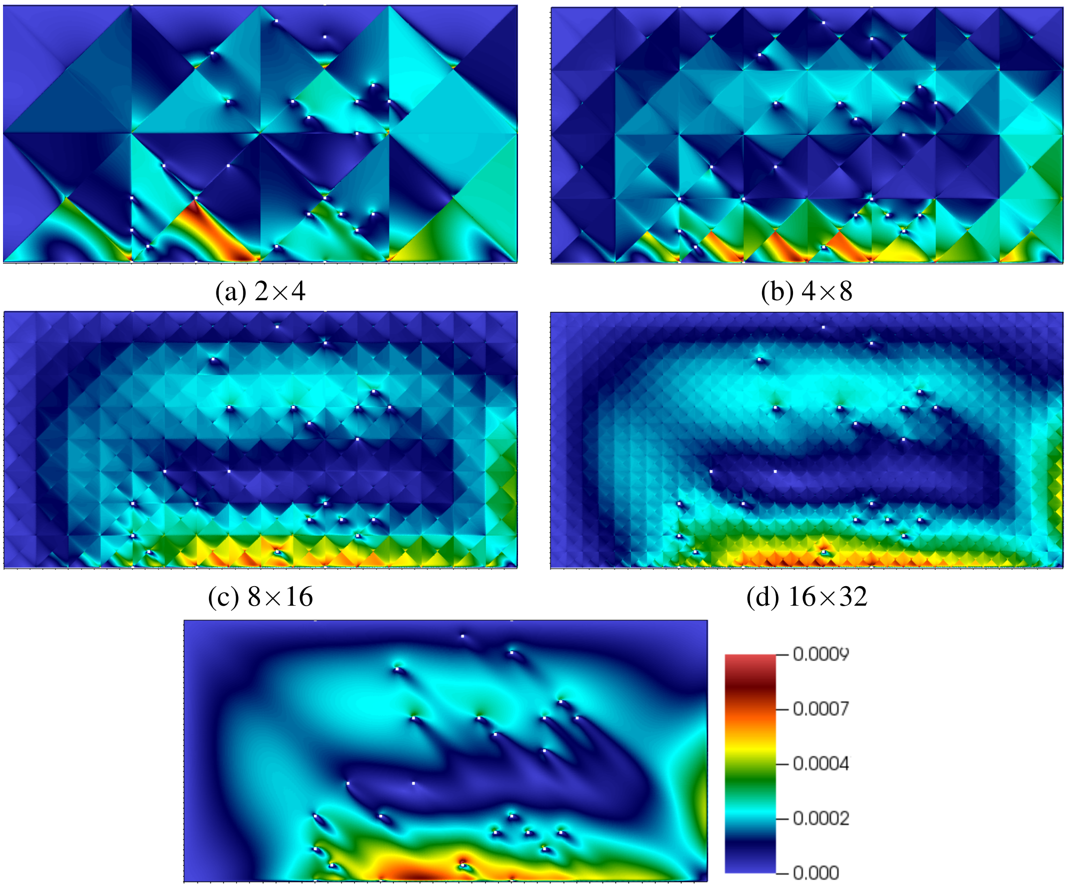
We recall that in multiscale finite element methods, multiscale basis functions are computed only once and then can be re-used multiple times for solving coarse-scale problems with different source terms or boundary conditions. The computation of matrices and the reconstruction of fine-scale solutions take very little time. Coarse-scale problems are not of too large sizes and can be solved rapidly even using only one processor. In conclusion, MsFEMs allow more important computational savings especially when multiple computations need to be performed on the same medium, with different boundary conditions or source terms.

6.4.2 Applications to Oseen flows

To take the convection term into account, we solve Oseen problems on the coarse mesh using Crouzeix-Raviart MsFEMs presented in this thesis. The Oseen problem [69, 70, 132] is a linearization of the Navier-Stokes problem where the nonlinear convection term is replaced by a linear Oseen term.

TABLE 6.5: Velocity errors of Oseen flows in the case (A) of Figure 6.5

Config.	H/ε	L^2 Rel. U			
		CR2_Os	CR3_Os	CR4_Os	CR4_Os_high
2×4	50	0.593	0.265	0.241	0.201
4×8	25	0.368	0.163	0.140	0.106
8×16	12.5	0.216	0.094	0.079	0.052
16×32	6.25	0.140	0.059	0.038	0.028



(a) 2×4

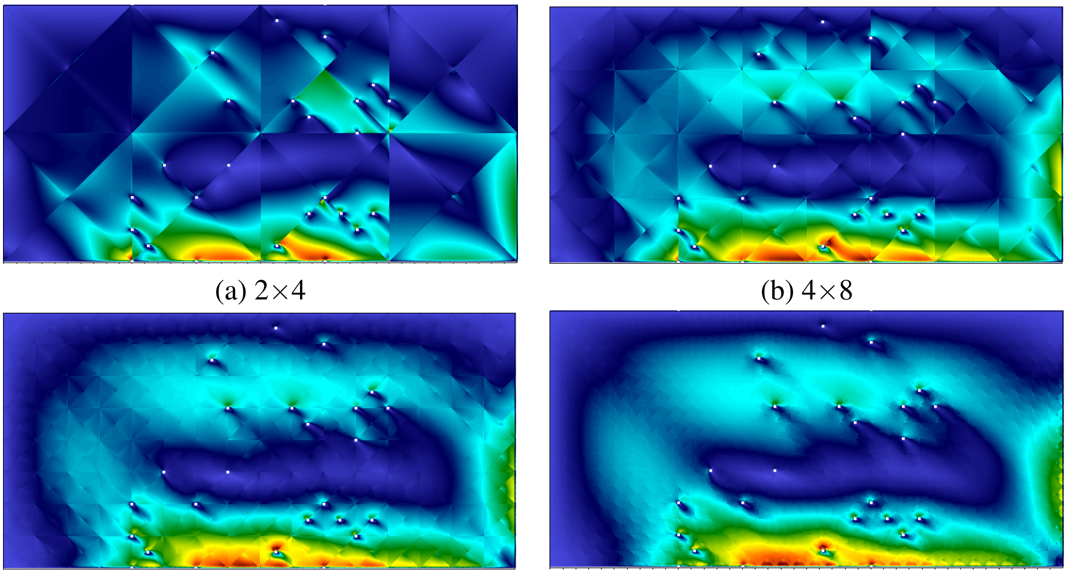
(b) 4×8

(c) 8×16

(d) 16×32

(e) reference solution

FIGURE 6.18: $|\boldsymbol{u}|$ contours of Oseen flows in the case (A) of Figure 6.5 computed by CR2_Os



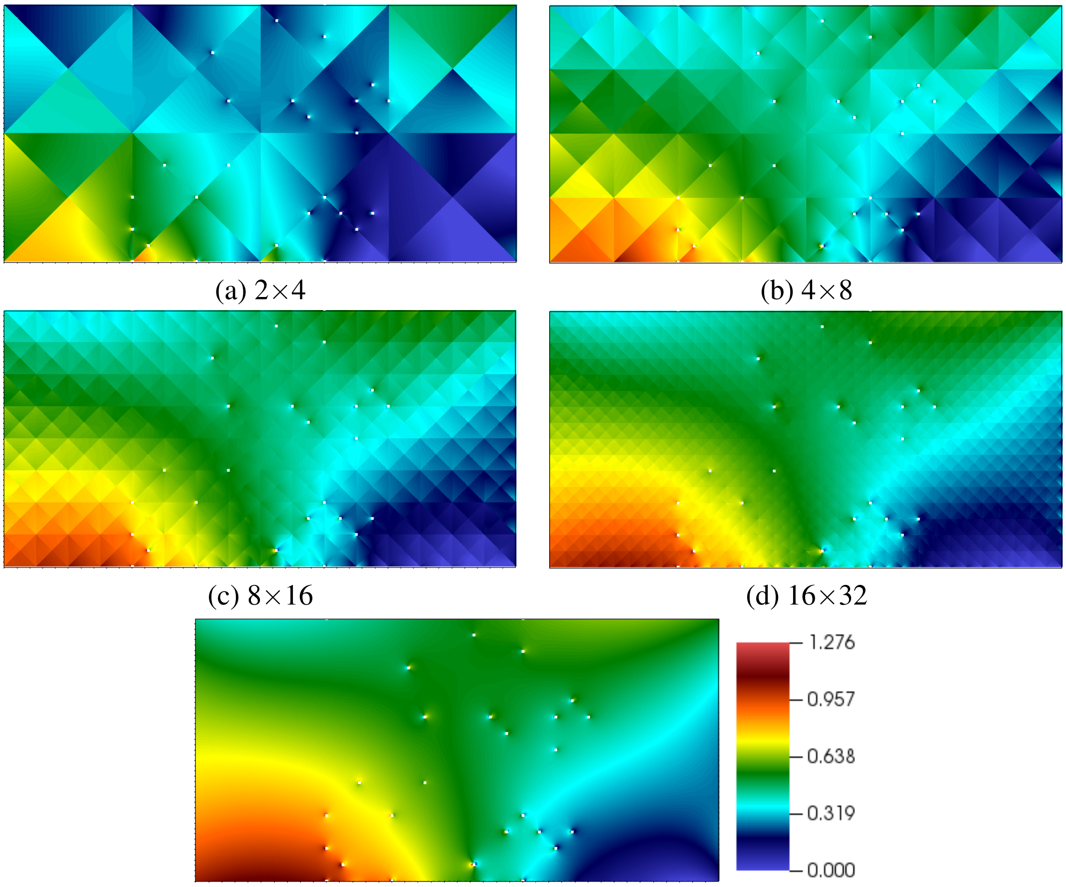
(a) 2×4

(b) 4×8

(c) 8×16

(d) 16×32

FIGURE 6.19: $|\boldsymbol{u}|$ contours of Oseen flows in the case (A) of Figure 6.5 computed by CR3_Os



(a) 2×4

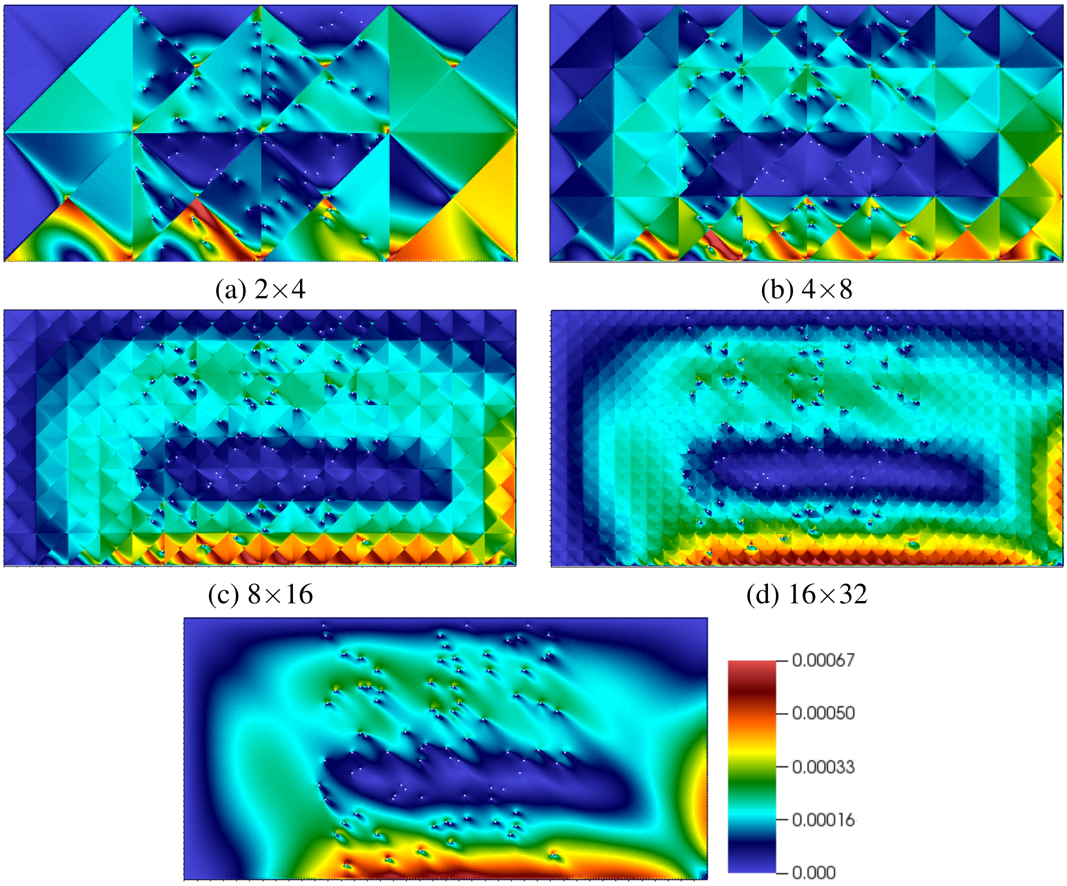
(b) 4×8

(c) 8×16

(d) 16×32

(e) reference

FIGURE 6.20: Pressure contours of Oseen flows in the case (A) of Figure 6.5 computed by CR3_Os

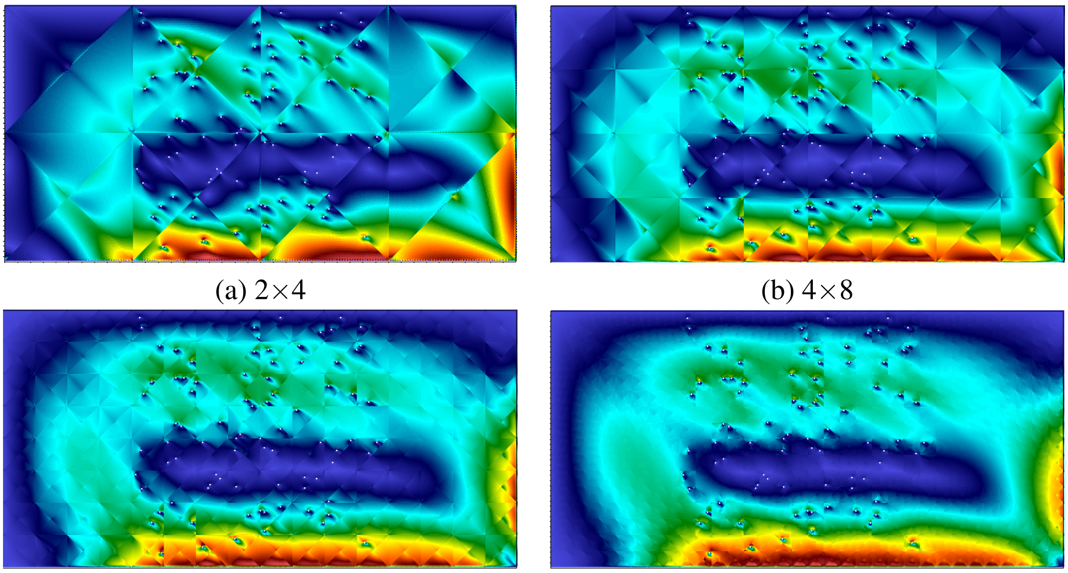


(a) 2×4 (b) 4×8

(c) 8×16 (d) 16×32

(e) reference solution

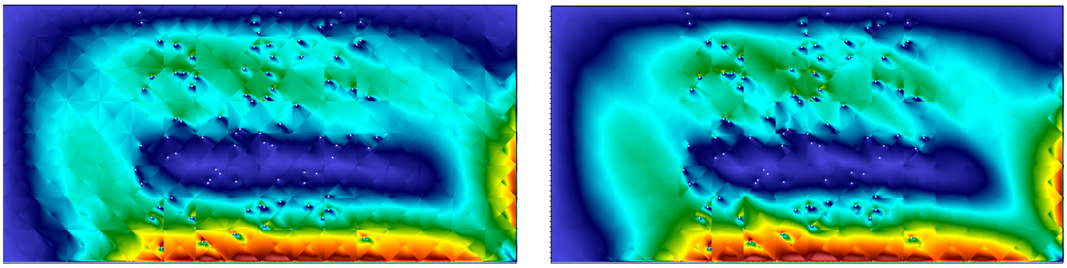
FIGURE 6.21: $|\boldsymbol{u}|$ contours of Oseen flows in the case (B) of Figure 6.5 computed by CR2_Os



(a) 2×4 (b) 4×8

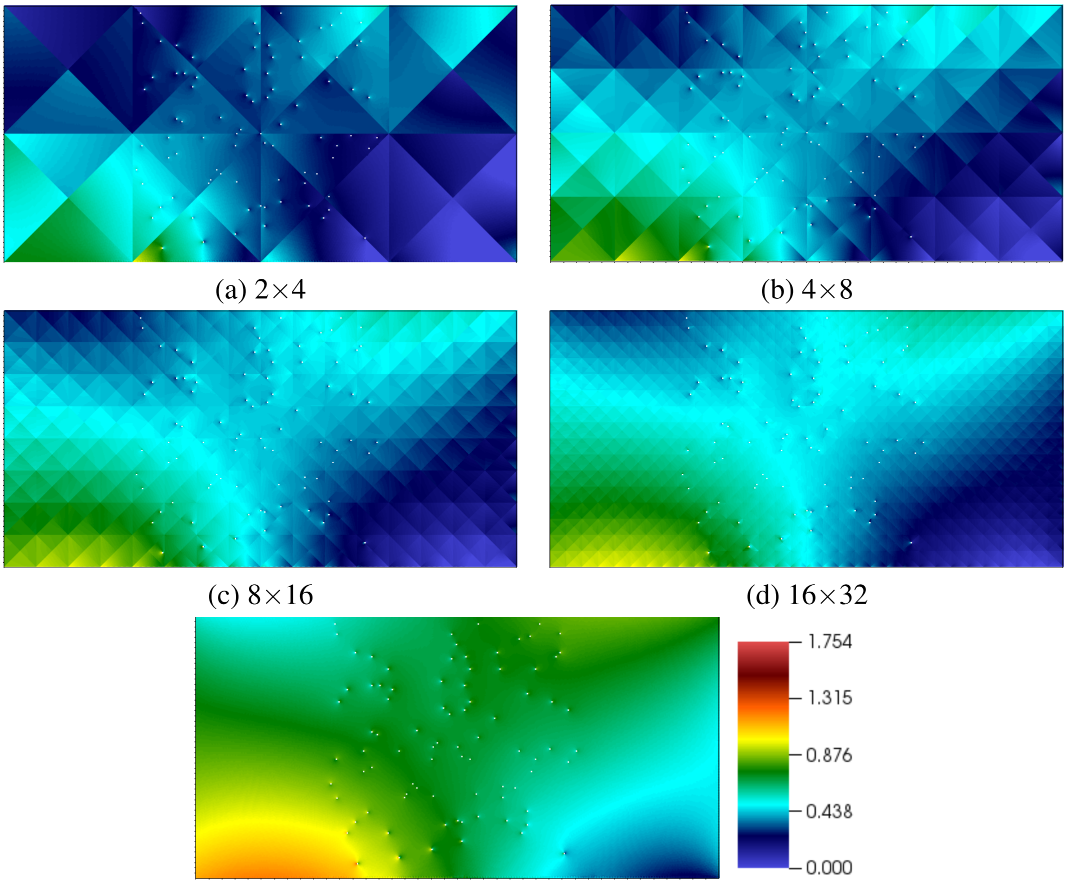
(c) 8×16 (d) 16×32

FIGURE 6.22: $|\boldsymbol{u}|$ contours of Oseen flows in the case (B) of Figure 6.5 with CR3_Os



(c) CR3_Os 8×16 (d) CR4_Os 8×16

FIGURE 6.23: $|\boldsymbol{u}|$ contours of Oseen flows in the case (B) of Figure 6.5 computed by CR3_Os and CR4_Os



(a) 2×4 (b) 4×8

(c) 8×16 (d) 16×32

(e) reference

FIGURE 6.24: Pressure contours of Oseen flows in the case (B) of Figure 6.5 computed by CR3_Os

6.4.2.1 Numerical convergence of Crouzeix-Raviart MsFEMs with respect to H

TABLE 6.6: Pressure errors of Oseen flows in the case (A) of Figure 6.5

Config.	H/ε	L^2 Rel. P			
		CR2_Os	CR3_Os	CR4_Os	CR4_Os_high
2×4	50	0.481	0.388	0.306	0.241
4×8	25	0.259	0.209	0.190	0.139
8×16	12.5	0.202	0.165	0.149	0.092
16×32	6.25	0.109	0.105	0.090	0.058

TABLE 6.7: Velocity errors of Oseen flows in the case (B) of Figure 6.5

Config.	H/ε	L^2 Rel. U			
		CR2_Os	CR3_Os	CR4_Os	CR4_Os_high
2×4	125	0.615	0.280	0.253	0.217
4×8	62.5	0.372	0.168	0.151	0.115
8×16	31.25	0.222	0.104	0.091	0.068
16×32	15.63	0.146	0.079	0.066	0.048

TABLE 6.8: Pressure errors of Oseen flows in the case (B) of Figure 6.5

Config.	H/ε	L^2 Rel. P			
		CR2_Os	CR3_Os	CR4_Os	CR4_Os_high
2×4	125	0.589	0.536	0.498	0.421
4×8	62.5	0.467	0.426	0.415	0.342
8×16	31.25	0.396	0.368	0.354	0.296
16×32	15.63	0.195	0.173	0.161	0.122

In both cases (A) and (B) of Figure 6.5, we solve Oseen problems on the coarse mesh using different Crouzeix-Raviart MsFEMs. The Oseen velocity is fixed to be $\boldsymbol{U}_o = (400, -400)$ and the source term is $\boldsymbol{f} = (1/2 - y, x - 1/2)$. It is important to note that the same Oseen velocity is used in local problems defined by Oseen equations. We impose $\boldsymbol{u} = \boldsymbol{0}$ on $\partial\Omega$. We fix the element size h of the fine meshes while decreasing the element size H of the coarse mesh. Reference solutions are computed on the same reference meshes as in subsection 6.4.1. Since Oseen problems are not available in *TrioCFD*, we implemented Oseen problems based on the Navier-Stokes problems implemented in the software.

Figures 6.18 to 6.19 and Figures 6.21 to 6.22 show the converging behavior of velocity solved by CR2_Os and CR3_Os in both cases (A) and (B) of Figure 6.5. It is shown that the velocity computed by CR3_Os is remarkably more accurate than that computed by CR2_Os. Important flow features are already well captured by CR3_Os on the coarse mesh 8×16 at a rather large ratio H/ε. Figure 6.23 shows that CR4_Os is slightly more accurate than CR3_Os. Figure 6.24 shows the convergence of the pressure solved by CR3_Os toward the reference solution in the case (B) of Figure 6.5. We observe that most important features of the pressure are well captured at the coarse mesh 16×32.

Tables 6.5 to 6.6, Tables 6.7 to 6.8 present the velocity and pressure errors in both cases (A) and (B) of Figure 6.5. It is shown that CR4_Os_high gives the most accurate velocity and pressure. In particular, CR4_Os_high improves significantly the accuracy of pressure compared to CR4_Os and CR3_Os. This underlines the usefulness of enriching the approximation space of pressure in order to improve the accuracy of both the velocity and pressure. Figure 6.25 and Figure 6.26 show the convergence of different Crouzeix-Raviart MsFEMs. It is shown that these MsFEMs have almost the same convergence rate but CR4_Os_high has the smallest errors in both case (A) and (B). The same phenomenon has also been observed in the high-order method proposed in [50]. In this thesis, only CR4 high with $n = 1$ has been implemented and tested. In the future work, an error analysis could be performed for the high-order method in order to find the relation between error of the method and the degree of polynomials n. It would be also interesting to implement the high-order MsFEMs for $n = 2$, $n = 3$ and so on and compare its performance with other Crouzeix-Raviart MsFEMs.

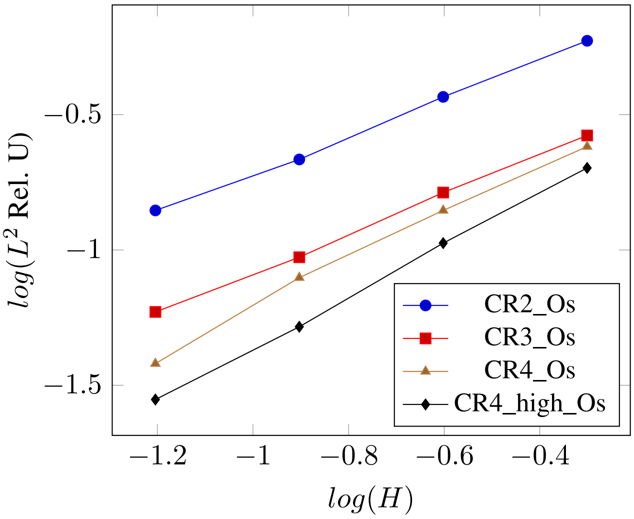


FIGURE 6.25: Numerical convergence of MsFEMs for Oseen flows in the case (A) of Figure 6.5

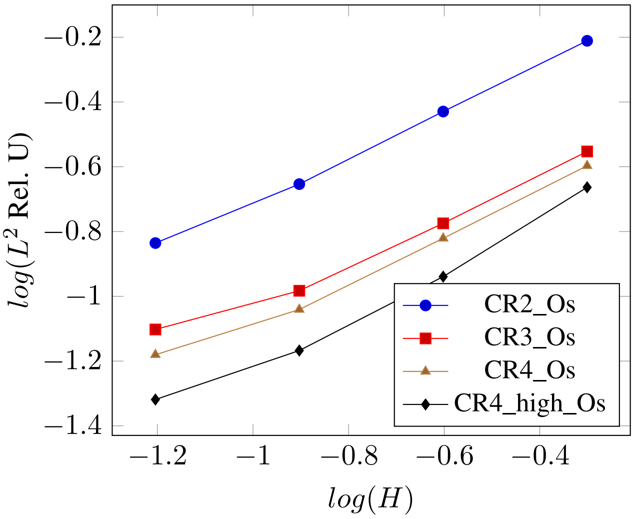


FIGURE 6.26: Numerical convergence of MsFEMs for Oseen flows in the case (B) of Figure 6.5

6.4.2.2 Error analysis with respect to the Reynolds number

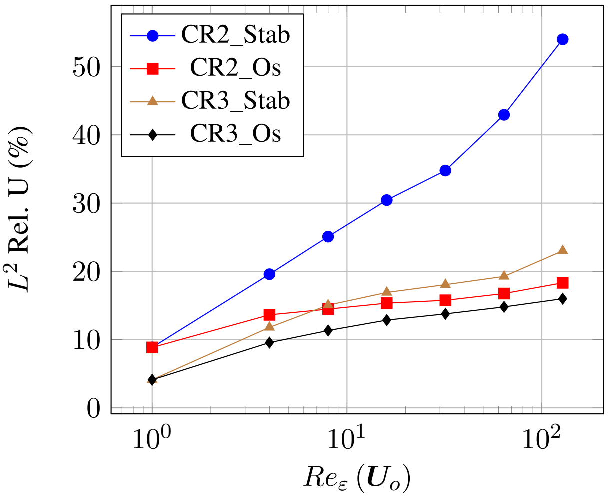


FIGURE 6.27: Velocity errors with respect to $Re_\varepsilon(\boldsymbol{U}_o)$ in the case (B) of Figure 6.5

In this numerical experiment, we solve Oseen flows in the case (B) of Figure 6.5 at different Oseen velocities $\boldsymbol{U}_o$. We impose a parabolic velocity $\boldsymbol{u} = U_{inlet} \times 4y(1-y)\,\boldsymbol{e}_1$ at the inlet and $\nabla \boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{0}$ at the outlet. The no-slip condition is imposed on other boundaries.

We define the fine- and coarse-scale Reynolds number for Oseen flows respectively by

$$Re_\varepsilon(\boldsymbol{U}_o) = \rho|\boldsymbol{U}_o|\varepsilon/\mu$$
$$Re_H(\boldsymbol{U}_o) = \rho|\boldsymbol{U}_o|L/\mu$$

with $\rho = 1$, $\mu = 1$ and L is the characteristic length of the computational domain. It is easy to see that $Re_\varepsilon(\boldsymbol{U}_o) \ll Re_H(\boldsymbol{U}_o)$ when $\varepsilon \ll L$.

In this numerical experiment, the Oseen velocity $\boldsymbol{U}_o$ is chosen as $\boldsymbol{U}_o = U_{inlet}\boldsymbol{e}_1$. We vary the value of U_{inlet} to change $\boldsymbol{U}_o$ and thus $Re_\varepsilon(\boldsymbol{U}_o)$. The same Oseen velocity is used in both local and coarse-scale Oseen problems. Thus each time the Oseen velocity $\boldsymbol{U}_o$ changes, we solve local Oseen problems again with the new value of $\boldsymbol{U}_o$ and obtain new multiscale basis functions.

We apply CR2, CR3, CR2_Os and CR3_Os to solve Oseen problems on the coarse mesh 8×16. We recall that local problems in CR2 and CR3 are defined by Stokes equations while local problems in CR2_Os and CR3_Os are defined by Oseen equations. In practice, when using CR2 and CR3, oscillations appeared for large values of $\boldsymbol{U}_o$. Thus the stabilized version CR2_Stab and CR3_Stab are used instead of CR2 and CR3. It is shown in Figure 6.27 that the CR2_Stab is quite sensitive to $Re_\varepsilon(\boldsymbol{U}_o)$ since velocity errors increase rapidly with respect to Reynolds numbers. However, CR3_Stab, CR2_Os and CR3_Os are more accurate and more robust in the sens that errors increase more slowly with respect to $Re_\varepsilon(\boldsymbol{U}_o)$. The most accurate results are given by CR3_Os.

Interestingly, we observe that the accuracy of CR3_Stab is very close to that of CR2_Os. We recall that in CR3_Stab, multiscale basis functions are defined by Stokes equations with weighting functions in the case of $s = 3$ defined by (6.1). Consequently, our result highlights that when using weighting functions in the case of $s = 3$ or $s = 4$ defined by (6.1), basis functions defined by Stokes equations can also solve Oseen problems rather accurately on the coarse mesh. It would be interesting to check if by using high-order Crouzeix-Raviart multiscale methods with basis functions defined by Stokes or Oseen equations, we can solve Navier-Stokes equations more correctly at even larger Reynolds numbers on the coarse mesh.

6.4.2.3 The MsFEM defined by adding solutions of local Stokes and Oseen problems

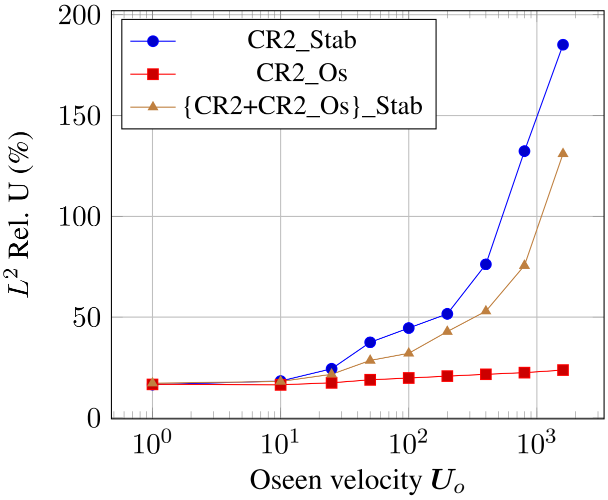


FIGURE 6.28: Velocity errors with respect to $\boldsymbol{U}_o$ in the case (B) of Figure 6.5

In the case (B) of Figure 6.5, we impose $\boldsymbol{u} = \mathbf{0}$ on $\partial\Omega$ and source term $\boldsymbol{f} = \left(\frac{1}{2} - y, x - \frac{1}{2}\right)$. In this numerical experiment, the Oseen velocity $\boldsymbol{U}_o$ varies from $(0,0)$ to $(1600, -1600)$. We solve Oseen problems on the coarse-mesh using CR2, CR2_Os and {CR2+CR2_Os}. Note that in CR2_Os, the same $\boldsymbol{U}_o$ is used in both local and coarse-scale Oseen problems. In order to preclude oscillations in the solution, stabilized methods CR2_stab and {CR2+CR2_Os}_Stab are used instead of CR2 and {CR2+CR2_Os}. No stabilization technique is necessary for CR2_Os since no oscillation appears in the solution.

As shown in Figure 6.28, when $\boldsymbol{U}_o$ is small, CR2_Stab, CR2_Os and {CR2+CR2_Os}_Stab yield very similar results. When $\boldsymbol{U}_o$ is larger than (100,-100), CR2_Os has much smaller velocity errors compared to CR2_Stab and {CR2+CR2_Os}_Stab. {CR2+CR2_Os}_Stab is slightly more accurate than the CR2_Stab but unfortunately much less accurate than CR2_Os.

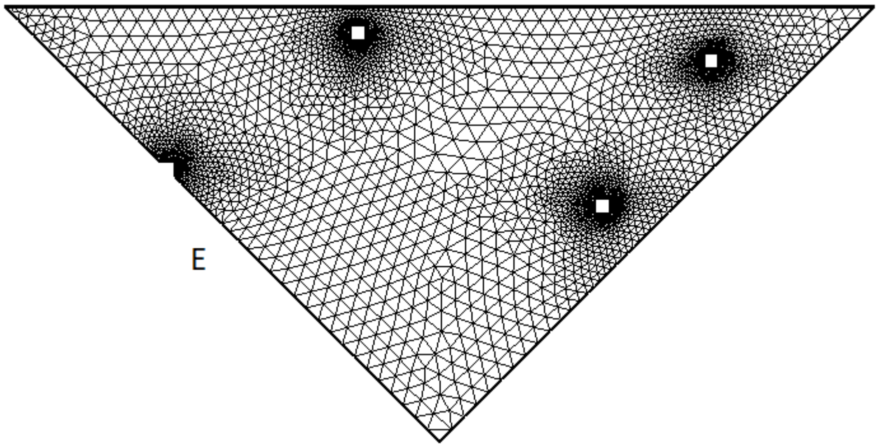
We conclude that when the convection is not dominant, these MsFEMs yield similar results. However, when the convection dominates, CR2_Os shows its advantage over CR2_Stab and {CR2+CR2_Os}_Stab. Contrary to expectation, the enriched method {CR2+CR2_Os}_Stab is more accurate than CR2_Stab but unfortunately much less accurate than CR2_Os. The reasons for this result is not entirely understood. One possible explanation is that we have only enriched the velocity approximation space, but not the pressure approximation space.

6.4.2.4 Multiscale basis functions defined by Oseen equations

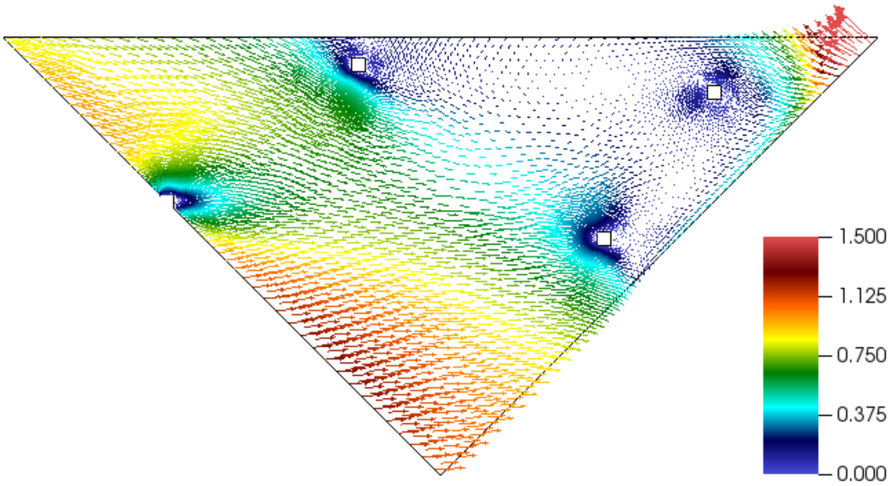
Figures 6.29 to 6.30 show multiscale basis functions computed on the same coarse element as shown in Figure 6.16 (A). The multiscale basis functions are computed respectively by CR2_Os and CR3_Os for a fixed Oseen velocity $\boldsymbol{U}_o = (200, 0)$. Comparing with basis functions shown in Figures 6.16 to 6.17, we remark that CR2_Os and CR3_Os capture correctly the inertial effects related to the Oseen term. Figures 6.29 to 6.30 reveal that multiscale basis functions capture successfully the presence of obstacles and no-slip boundary condition is well imposed on obstacles. Even on the boudnary E cut by obstacles, basis functions can still successfully satisfy $\int_E \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{\omega}_{E,j} = \delta_{ij}$ for $i, j = 1, \cdots, 3$ defined in local problems.

6.4.3 Applications to Navier-Stokes flows

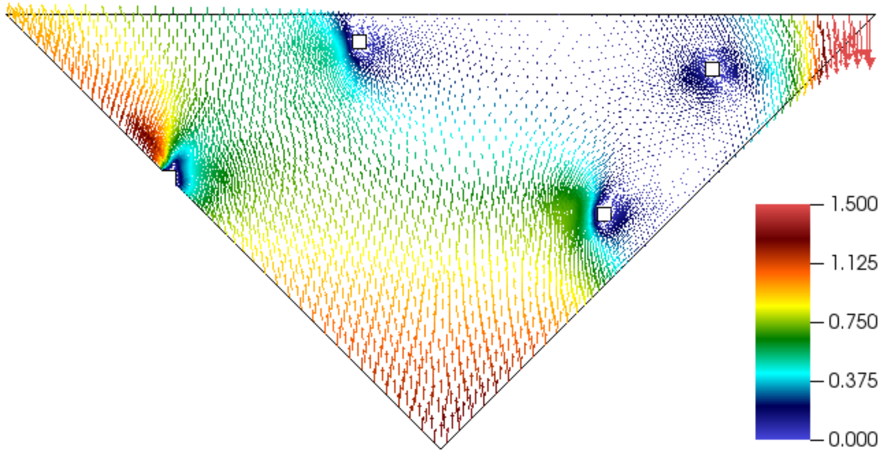
Now we solve the nonlinear Navier-Stokes flows on the coarse mesh in both cases (A) and (B). Numerical simulations are carried out on a fixed coarse mesh 8×16. The reference solutions



(A) The coarse element with the boundary E

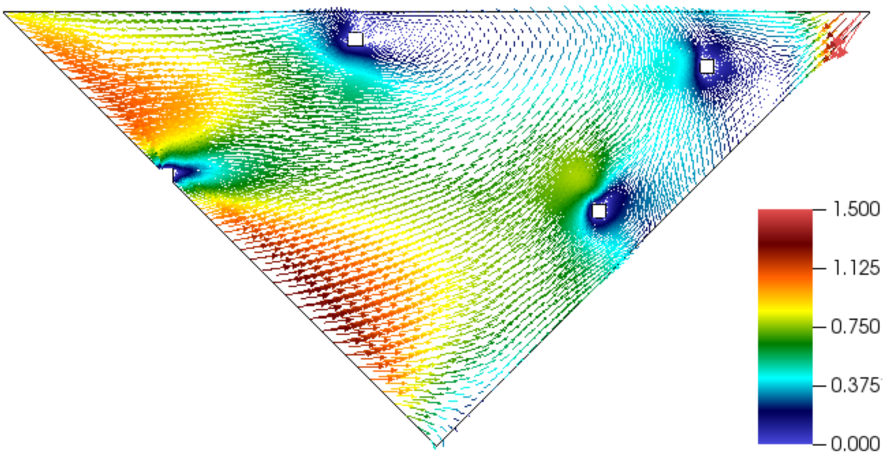


(B) Vector $\mathbf{\Phi}_{E,1}$

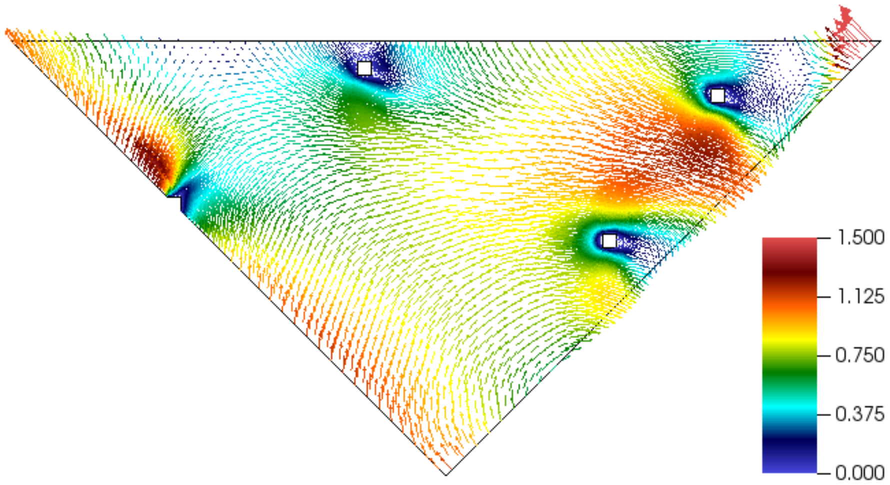


(C) Vector $\mathbf{\Phi}_{E,2}$

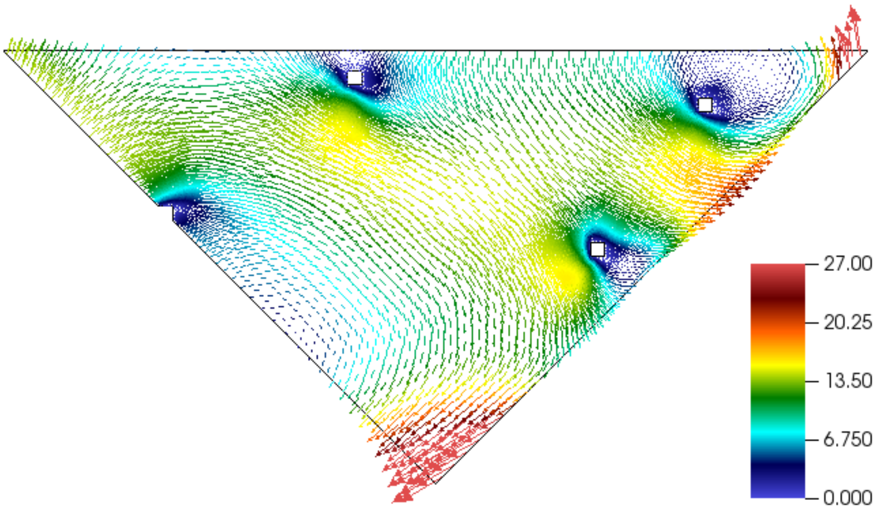
FIGURE 6.29: The coarse element and multiscale basis functions computed by CR2_Os



(A) Vector $\mathbf{\Phi}_{E,1}$



(B) Vector $\mathbf{\Phi}_{E,2}$



(C) Vector $\mathbf{\Phi}_{E,3}$

FIGURE 6.30: Multiscale basis functions computed by CR3

are discretized by the finite volume element method and solved by the *prediction-correction* algorithm in *TrioCFD*. Besides, the Upwind scheme [65] is used to stabilize the convection term of Navier-Stokes problems.

We impose a parabolic velocity $\boldsymbol{u} = U_{inlet} \times 4y\,(1-y)\,\boldsymbol{e}_1$ at the inlet and $\nabla \boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{0}$ at the outlet. The no-slip condition is imposed on other boundaries. The fine- and coarse-scale Reynolds number are defined respectively by

$$Re_\varepsilon = \rho |U_{inlet}| \varepsilon / \mu$$
$$Re_H = \rho |U_{inlet}| L / \mu$$

with L the characteristic length of the domain. It is known that $Re_H \gg Re_\varepsilon$ when $L \gg \varepsilon$ in highly heterogeneous media. In this study, we have $Re_H \approx 67 \times Re_\varepsilon$ in case (A) and $Re_H \approx 333 \times Re_\varepsilon$ in case (B).

As presented in Chapter 3, we approximate the Oseen velocity in local Oseen problems of CR2_Os and CR3_Os in the following way. Firstly, we solve the Stokes problem on the coarse mesh by CR2 or CR3 and obtain the coarse-scale velocity. Secondly, we compute an average of the coarse-scale velocity on each coarse element K of the coarse mesh $\mathcal{T}_H$. The average on each coarse element K is denoted by $\boldsymbol{U}_o|_K$. Consequently, the Oseen velocity field $\boldsymbol{U}_o$ is in fact the vector $(\boldsymbol{U}_o|_K)_{K \in \mathcal{T}_H}$. Thus the Oseen velocity field $\boldsymbol{U}_o$ is constant on each coarse element. This velocity field is used in the definition of local Oseen problems of CR2_Os or CR3_Os.

It should be noted that the approximation of the Oseen velocity field should be close to the convective velocity field in Navier-Stokes problems in order to improve the accuracy of CR2_Os and CR3_Os. There are other methods to get a better approximation of the Oseen velocity in local Oseen problems. For example, instead of solving Stokes problems, we can solve Navier-Stokes problems on the coarse mesh by CR2 or CR3. But the SUPG stabilization technique is needed if the convection dominates in Navier-Stokes problems.

After obtaining the Oseen velocity field $\boldsymbol{U}_o$ which is constant on each coarse element, we solve local Oseen problems of CR2_Os or CR3_Os and obtain multiscale basis functions. Then we solve Navier-Stokes problems using these basis functions on the coarse mesh. Figure 6.31 and Figure 6.32 show the velocity errors computed by various multiscale methods at different Re_ε in both cases (A) and (B). When the diffusion dominates, i.e. when $Re_\varepsilon < 5$ in the case (A) and $Re_\varepsilon < 2$ in the case (B), CR2_Stab and CR3_Stab have almost the same results as CR2_Os and CR3_Os.

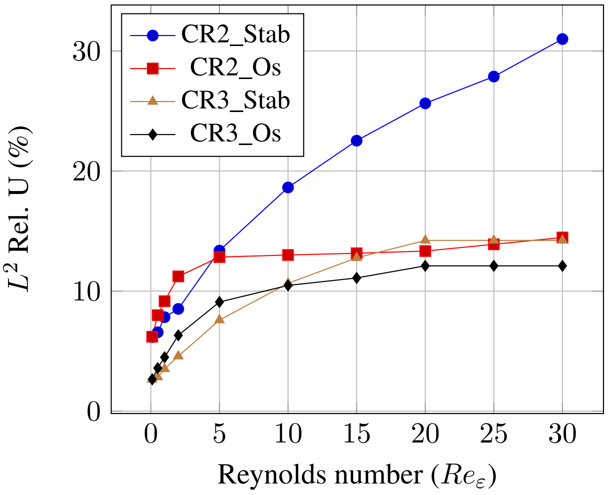


FIGURE 6.31: Velocity errors with respect to Re_ε in the case (A) of Figure 6.5

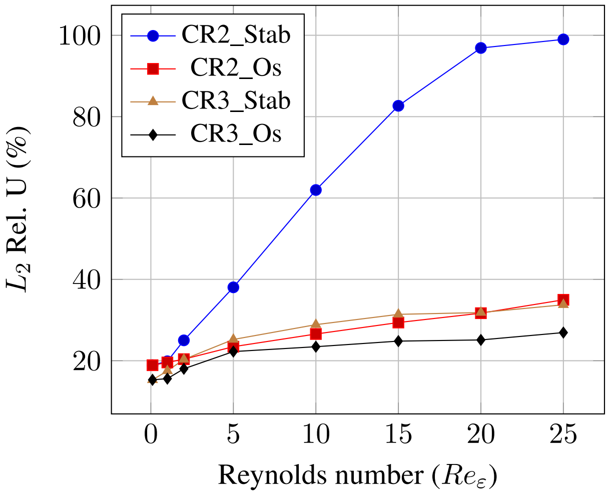


FIGURE 6.32: Velocity errors with respect to Re_ε in the case (B) of Figure 6.5

When the convection dominates, CR2_Os, CR3_Os and CR3_Stab are significantly more accurate and more robust with respect to Re_ε compared to CR2_Stab. it is shown that CR2_Stab is very sensitive to Re_ε and the velocity errors increase rapidly with respect to Re_ε. We observe that CR2_Os is much more accurate than CR2_Stab. This highlights that using the same weighting functions (the case of $s = 2$ defined by (6.1)), multiscale basis functions defined by Oseen equations are more accurate for solving Navier-Stokes problems on the coarse mesh. Interestingly, we also observe that CR2_Os and CR3_Stab have almost the same accuracy. This confirms that with weighting functions in the case of $s = 3$ or $s = 4$ defined by (6.1) or high-order multiscale methods such as CR4_high, multiscale basis functions defined by Stokes equations can also solve Navier-Stokes problems rather accurately on the coarse mesh.

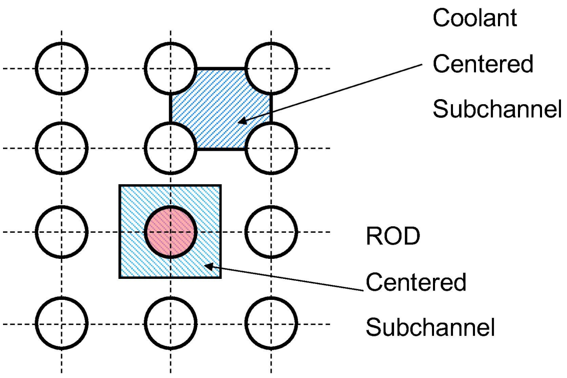
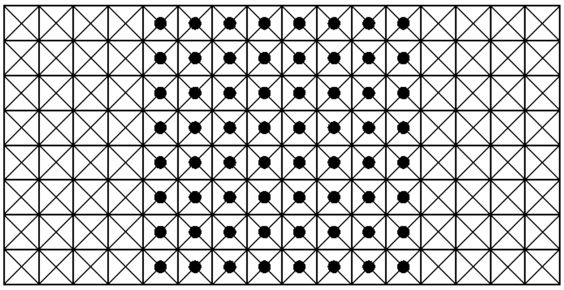


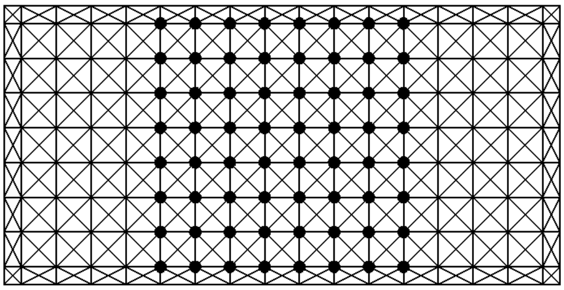
FIGURE 6.33: Definition of fuel assembly sub-channels (image from [115])

6.5 Simulations in two-dimensional periodic heterogeneous media

The case of flow through closely spaced periodic cylinder arrays is important for applications in the nuclear industry, such as flow around nuclear fuel rods and flow past heat exchange coils in steam generators. In this section, we apply MsFEMs to solve flow problems past periodic arrays of obstacles and show the performance of MsFEMs.



(A) Rod centered sub-channel



(B) Coolant centered sub-channel

FIGURE 6.34: Coarse meshes superposed on solid obstacles on partition A and partition B

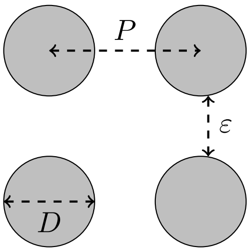


FIGURE 6.35: Periodic setting: the pitch P, the diameter D and the gap ε

TABLE 6.9: Error analysis of Stokes flows in a periodic heterogeneous medium

Partition	Config. H/ε	L^2 Rel. U			L^2 Rel. P		
		CR2	CR3	CR4	CR2	CR3	CR4
A	5.31	0.852	0.515	0.499	0.934	0.356	0.344
	2.65	0.823	0.421	0.470	0.927	0.304	0.291
	1.43	0.731	0.284	0.269	0.908	0.224	0.207
B	1.33	0.262	0.101	0.091	0.174	0.038	0.030

6.5.1 Numerical convergence of Crouzeix-Raviart MsFEMs with respect to H

In the nuclear industry, sub-channel thermal-hydraulic codes are widely used to estimate the thermal-hydraulic safety margins of the nuclear reactor core. A sub-channel is defined as a flow passage formed between fuel rods or between some fuel rods and wall of channel tube. As shown in Figure 6.33, the sub-channel can be formed by either coolant centered sub-channels or rod centered sub-channels. For more details about the sub-channel thermal-hydraulic analysis, the reader can refer to [115].

In this study, we consider a domain with a square array of 8×8 periodically placed obstacles. We impose a parabolic velocity $\boldsymbol{u} = 4y\,(1-y)\,\boldsymbol{e}_1$ at the inlet and $\nabla\boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{0}$ at the outlet. The no-slip condition is imposed on other boundaries. We fix the pitch to diameter ratio as $P/D = 1.33$, where the pitch P and D are shown in Figure 6.35. As shown in Figure 6.34, we consider in this study the partitions A and B which correspond respectively to the rod centered sub-channel and the coolant centered sub-channel. In the numerical convergence study, we fix the element size h of fine meshes and vary only the element size H of the coarse-scale mesh.

Table 6.9 shows the numerical convergence of CR2, CR3 and CR4 with respect to H. It is shown that for both velocity and pressure, CR4 is slightly more accurate than CR3 and they are both significantly more accurate than CR2. We observe that the velocity and pressure errors of CR2 remain quite large even on the mesh with $H/\varepsilon = 1.43$.

Now let us compare results computed on the partitions A and B. For CR2, the velocity error using the partition A is 0.731 whereas that of using the partition B is only 0.262. For CR3, the velocity error using the partition A is 0.291 whereas that of using the partition B is only 0.101. This highlights that both CR2 and CR3 are significantly more accurate on the partition B than the partition A.

Figure 6.36 shows the velocity computed respectively using the partitions A and B. Comparing Figure 6.36 (a) and (b), we observe that using the partition A, the solution of CR2 is almost wrong whereas that of CR3 is rather accurate. Comparing Figure 6.36 (c) and (d), we observe that using the partition B, interestingly, both CR2 and CR3 yield rather accurate results.

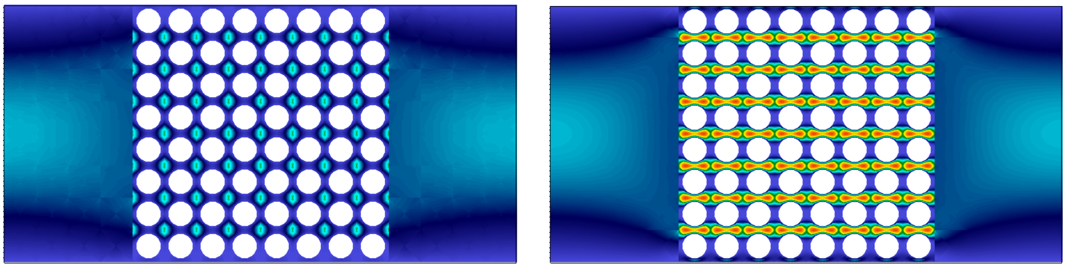
We conclude that: (i) The numerical convergence study in Table 6.9 shows that CR2 is not sufficiently accurate in periodic heterogeneous media and this result is consistent with the findings reported in [95]. It is necessary to use CR3, CR4 or high-order multiscale methods such as CR4_high to obtain much more accurate solutions. (ii) Both CR2 and CR3 can yield much more accurate results on the partition B than on the partition A. This result is interesting because it reveals that when using the partition B, CR2 can also give rather accurate solutions in periodic heterogeneous media.

6.5.2 Error analysis with respect to the heterogeneity

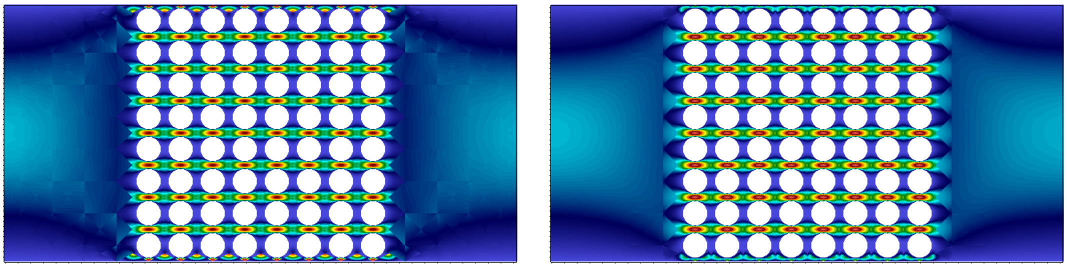
We apply CR3 to solve Navier-Stokes problems in a domain with a square array of periodically placed obstacles. A schematic description of the medium is shown in Figure 6.37 where obstacles are located in the center of the medium. The objective of this study is to see the performance of Crouzeix-Raviart MsFEMs when the computational domain becomes more and more heterogeneous. We impose a parabolic velocity at the inlet and $\nabla\boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{0}$ at the outlet. The no-slip condition is imposed on other boundaries.

We fix the element size H of the coarse mesh and increase the number of obstacles from 400 to 30,000 while decreasing their diameter D. As shown in Figure 6.35, when the ratio P/D is fixed, ε decreases with D. The element size h of fine meshes deceases also with ε and satisfies at least $\varepsilon/h > 5$ in order to well capture the presence of small obstacles.

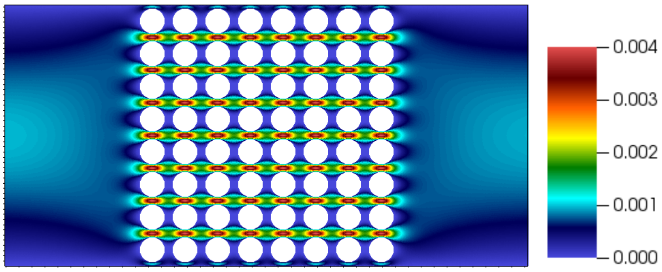
Note that each time we add more obstacles in the medium, the reference solution needs to be recomputed since the medium has changed. Making use of the periodicity of the medium, we use Algorithm 5.3 in Chapter 5 to generate necessary meshes and to compute multiscale basis



(a) Partition A with CR2 (b) Partition A with CR3



(c) Partition B with CR2 (d) Partition B with CR3



(e) reference solution

FIGURE 6.36: $|\boldsymbol{u}|$ computed with CR2 and CR3 on partition A and B

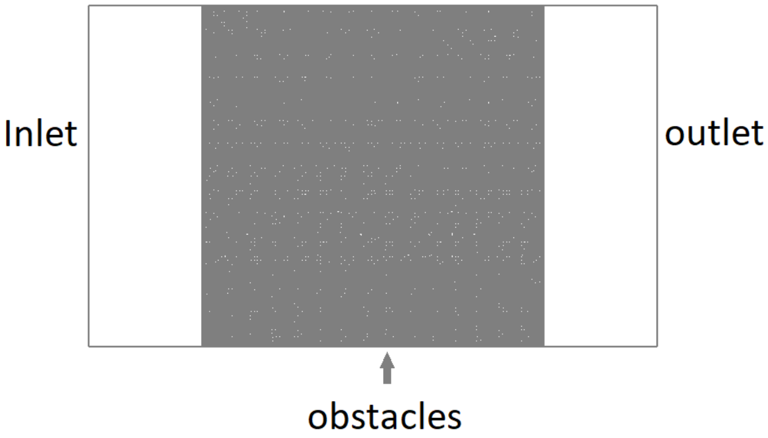


FIGURE 6.37: Schematic description of the medium containing periodic arrays of small obstacles (too many to show clearly)

functions. Finally, we need to construct multiscale basis functions only on the reference cell shown in Figure 6.39 and then we copy them to other fine meshes in the domain. Thus this algorithm allows great computational savings. However, it should be noted that the computation of reference solutions is very expensive in terms of time, especially when there are more than 10,000 obstacles in the domain. It can take more than 10 hours using 350 processors and the *prediction-correction* algorithm.

Figure 6.38 shows that both the velocity and pressure errors decrease with the number of obstacles in the medium. This result is coherent with the estimation (6.6). In the right hand side of (6.6), since H is fixed, when adding more obstacles and decreasing ε, the term $\sqrt{\frac{\varepsilon}{H}}$ decreases thus with ε.

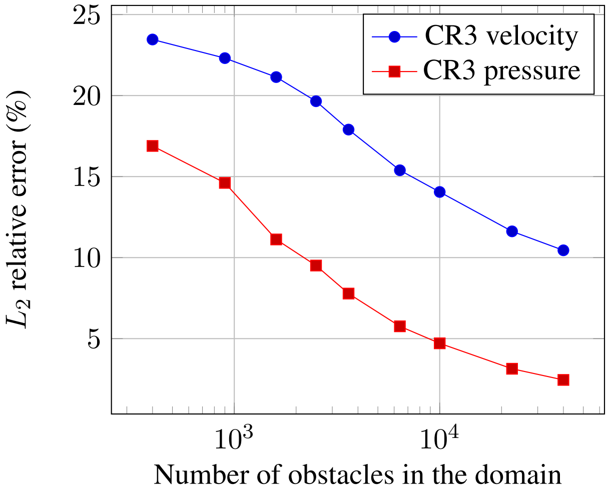


FIGURE 6.38: Velocity errors with respect to the number of obstacles on a fixed coarse mesh

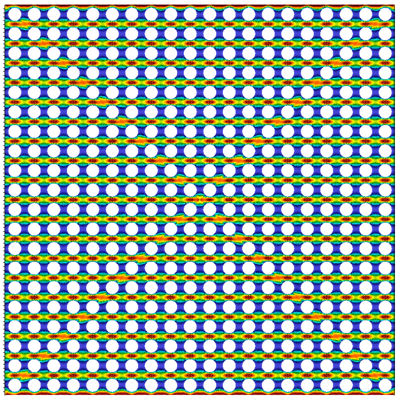


FIGURE 6.39: Velocity field $|\boldsymbol{u}|$ in the reference cell with 20×20 obstacles

Figure 6.39 reveals a zoom of the velocity contour in the reference cell containing 20×20 periodically placed obstacles. It is shown that important fine-scale features are correctly captured by Crouzeix-Raviart MsFEMs even when the domain is highly heterogeneous with numerous solids obstacles. This result is very important since it strengthens our conviction that Crouzeix-Raviart MsFEMs can be applied to solve flow problems successfully in periodic heterogeneous media, such as flows around fuel rods and flow past heat exchange coils in steam generators in the nuclear industry.

6.5.3 Applications to highly heterogeneous media

Now we present two demonstrative applications of Crouzeix-Raviart MsFEMs in highly heterogeneous media with the presence of numerous solids obstacles. It is very expensive in terms of computing time to solve flow problems in such heterogeneous media using classical numerical methods. In this situation, multiscale methods show the superior performance compared to classical numerical methods.

6.5.3.1 Flows through periodic arrays of assemblies with small defects

In this simulation, we solve Stokes problem by CR3 in a domain $\Omega = [0, 2.5] \times [0, 1.5]$ containing 90,000 densely-placed obstacles. A schematic description of the domain is shown in Figure 6.37 in which obstacles are located in the center of the medium. We impose a parabolic velocity at the inlet and $\nabla \boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{0}$ at the outlet. The no-slip condition is imposed on other boundaries.

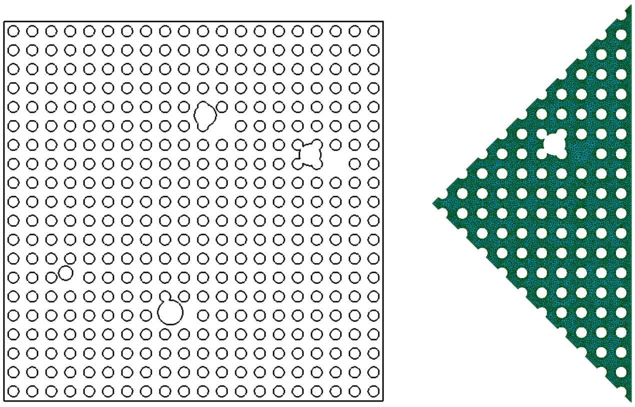
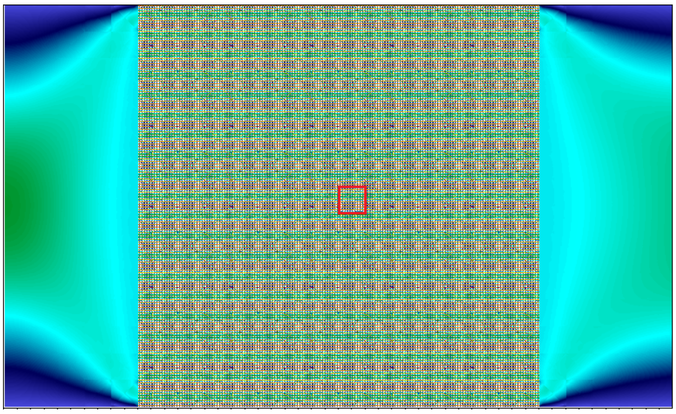


FIGURE 6.40: The reference cell containing periodically placed obstacles (left) and one coarse element (right)

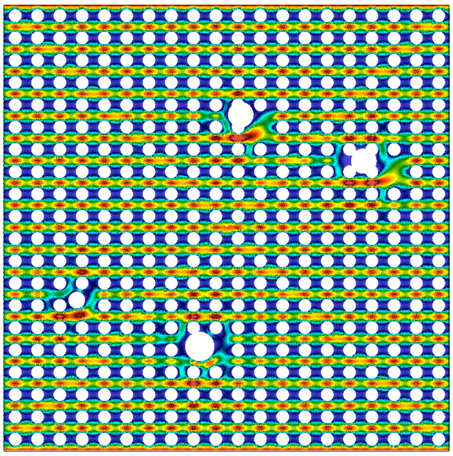
The part of the medium containing obstacles is formed by a periodic array of the reference cell shown in the left of Figure 6.40. The reference cell can be considered as a fuel assembly containing about 20 × 20 periodically placed fuel rods but with some small defects. These defects may exist when some fuel rods are deformed in accidental operations. Making use of the periodicity of the medium, we use Algorithm 5.3 in Chapter 5 to generate necessary meshes and to compute multiscale basis functions. The reference cell is cut into four triangular coarse elements and one of them is shown in Figure 6.40. We need to solve local problems only on the four coarse elements and then copy the basis functions to other coarse elements.

The coarse mesh consists of 1500 coarse elements and each coarse element has about 70,000 small elements in average. As a result, there are about 10^8 small elements in the entire domain Ω. Thus it is very expensive to solve flow problems in such a domain using classical numerical methods. However, using the *SALOME-TrioCFD-VisIt* multiscale simulation chain, the total computing time is about 2 hours. In particular, about 80% of the total time is taken by the solution of local problems using the modified *prediction-correction* algorithm (see Chapter 4). When a coarse element contains more obstacles, local problems are of larger sizes and the solution of local problems is more time-consuming.

As mentioned previously, the optimal performance of the *prediction-correction* algorithm in *TrioCFD* is when each processor treats between 20,000 and 30,000 mesh elements. Thus it is not efficient to solve local problems on a fine mesh containing 60,000 elements. Two processors should be used to have the optimal performance. However, we have not developed the intra-cellular parallelism (see section 5.1) for the solution of local problems in this thesis. In the end of the thesis, we implemented a direct solver which can solve local problems more rapidly and can greatly reduce the computing time taken by the solution of local problems.



(A) Velocity field $|\boldsymbol{u}|$ in the entire domain



(B) A zoom

FIGURE 6.41: Velocity field $|\boldsymbol{u}|$ of Stokes flows past 90,000 obstacles

Figure 6.41 shows the velocity solution in the entire domain Ω as well as a zoom of velocity on the reference cell. We see that fine-scale flow features and the presence of densely-placed obstacles are well captured by the multiscale method CR3.

6.5.3.2 Flows through periodic arrays of arbitrary obstacles

Now we solve Stokes problems with CR3 in a heterogeneous domain $\Omega = [0, 3.2] \times [0, 1.6]$ containing 16,384 densely-placed small obstacles. A schematic representation of the medium is shown in Figure 6.37 where obstacles are located in the center of the medium. The medium can be considered as a periodic array of the reference cell shown in the left of Figure 6.42. The reference cell contains 63 randomly placed obstacles. This medium can be considered as a good two-dimensional approximation of the core of the Pebble Bed Reactor (see Figure 1.2) which contains numerous arbitrarily placed spherical fuel elements and moderator elements.

Making use of the periodicity of the medium, we use Algorithm 5.3 in Chapter 5 to create necessary meshes and to compute multiscale basis functions. The reference cell is cut into four coarse elements and one of them is shown in the right of Figure 6.42. We need to solve local problems only in the four coarse elements and then copy them to other coarse elements.

In this heterogeneous medium, we impose a parabolic velocity at the inlet and $\nabla \boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{0}$ at the outlet. The no-slip condition is imposed on other boundaries. The coarse mesh consists of

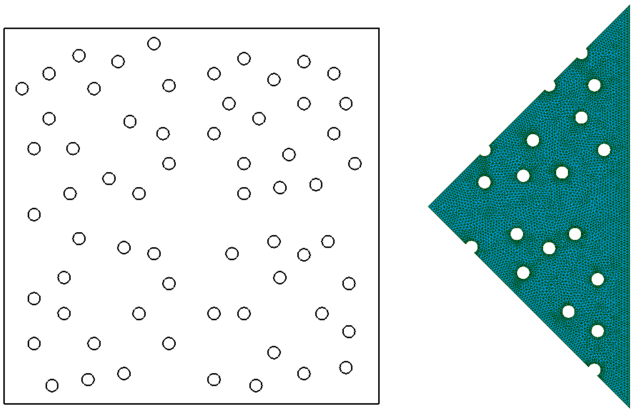


FIGURE 6.42: The reference cell containing arbitrarily placed obstacles (left) and one coarse element (right)

2048 coarse elements and each coarse element is made of about 60,000 small elements in average. In total, there are about 1×10^8 small elements in the global domain. The total computing time taken by the *SALOME-TrioCFD-VisIt* multiscale simulation chain is about 2 hours and the solution of local problems takes 90% of the total time. Note that local problems are solved by the *prediction-correction* algorithm (see Chapter 4).

Figure 6.43 shows the velocity field in the entire domain and a zoom of the velocity in a particular region. It is shown that important fine-scale flow features and the presence of densely placed small obstacles are very well captured by CR3.

To summarize, we have presented two demonstrative applications of Crouzeix-Raviart MsFEMs in highly heterogeneous media. It is very expensive to apply classical numerical methods to solve flow problems in such media whereas Crouzeix-Raviart MsFEMs yield rather satisfying results at reasonable costs.

6.6 Simulations in three-dimensional media

Now we apply Crouzeix-Raviart MsFEMs to solve flow problems in three-dimensional media. All the numerical simulations presented in this section are performed with CR3. The objective is to validate and show the good performance of Crouzeix-Raviart MsFEMs in three dimensional heterogeneous media. Besides, due to limited computing resources, no reference solutions are computed by *TrioCFD* and thus no error analysis is made.

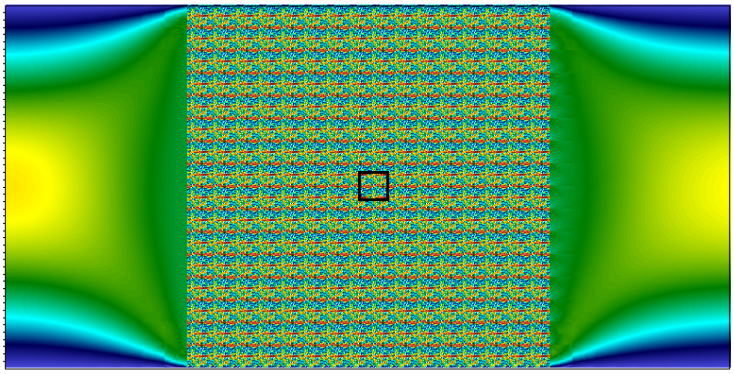
6.6.1 Flows in a homogeneous medium

We solve Stokes problems with CR3 in a three-dimensional homogeneous domain $\Omega = [0,1] \times [0,1] \times [0,1]$. We impose a parabolic velocity $\boldsymbol{u} = xy(1.0 - x)(1.0 - y)\boldsymbol{e}_z$ at the inlet and $\nabla \boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{0}$ at the outlet. The no-slip boundary condition is imposed on other boundaries. We recall that $\{\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z\}$ is the canonical basis of $\mathbb{R}^3$.

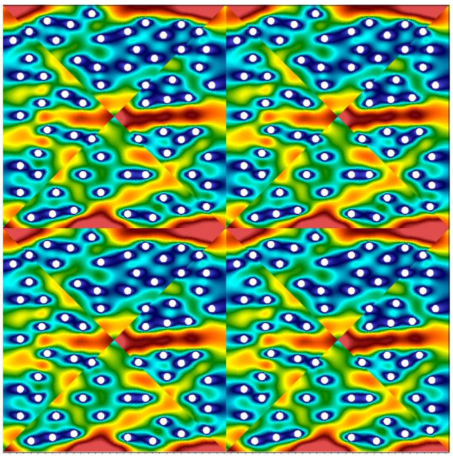
Figure 6.44 (A) and (B) show respectively the parabolic velocity profile imposed at the inlet and the velocity contour at the plane $x = 0.5$. It is shown that the parabolic velocity profile is correctly imposed and the velocity field agrees well with theoretical expectations.

6.6.2 Applications to a non-periodic heterogeneous medium

We solve a Stokes problem in a domain $\Omega = [0,1] \times [0,1] \times [0,2]$ with some randomly placed obstacles. As shown in Figure 6.45 (A), cylindrical obstacles are placed horizontally parallel

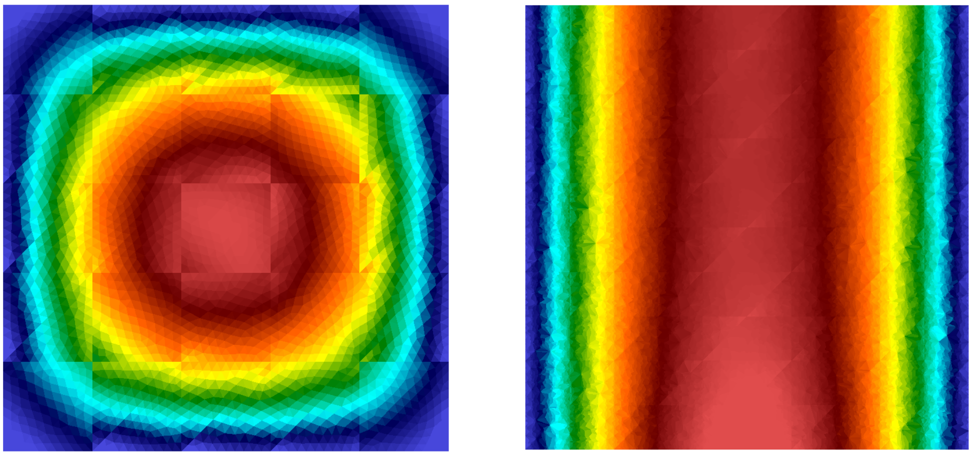


(A) Velocity field $|\boldsymbol{u}|$ in the entire domain



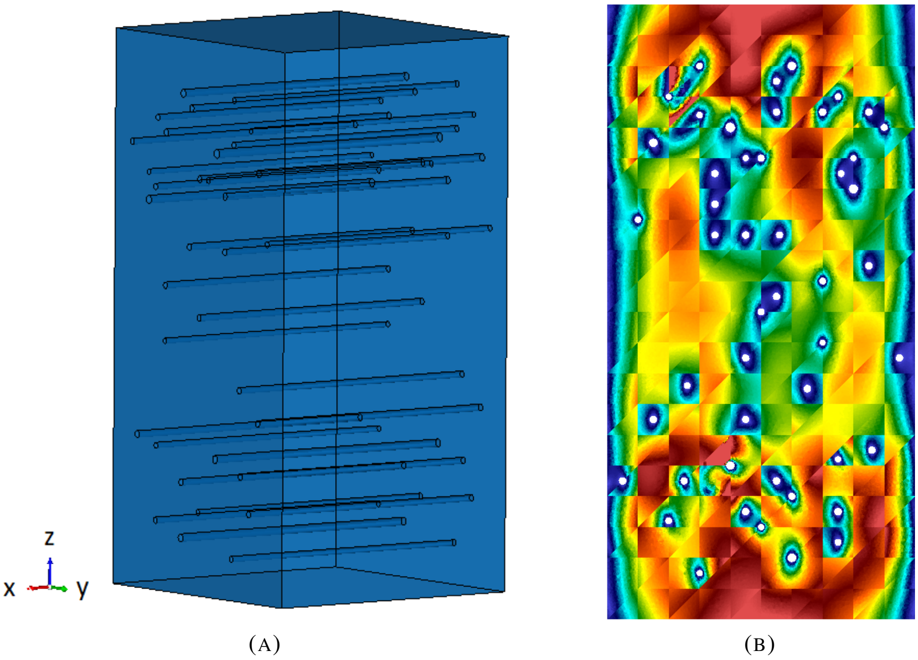
(B) A zoom

FIGURE 6.43: Velocity field $|\boldsymbol{u}|$ of Stokes flows past randomly placed obstacles



(A) Inlet velocity (B) Slice at $x = 0.5$

FIGURE 6.44: Velocity field $|\boldsymbol{u}|$ computed by CR3 in a homogeneous medium



(A) (B)

FIGURE 6.45: Computational domain (A) and a slice of velocity at $x = 0.5$ (B)

to X-axis in the domain. The diameter of cylinders is between $\varepsilon = 0.02$ and $\varepsilon = 0.032$. We impose a parabolic velocity $\boldsymbol{u} = xy(1.0 - x)(1.0 - y)\boldsymbol{e}_z$ at the inlet and boundary condition $\nabla \boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \mathbf{0}$ at the outlet. The no-slip boundary condition is imposed on other boundaries.

The coarse mesh consists of 1500 coarse elements and each coarse elements contain about 80,000 small elements. There are more than 1×10^8 small elements in the entire domain. No reference solution is solved due to limited computing resources in this thesis. Note that the multiscale basis functions are stored in the SAUV format and these files have taken a storage space of 20 Go on the local disk.

Figure 6.45 shows a vertical slice of the velocity field at the plane $x = 0.5$. It is shown that macroscopic flow features as well as the presence of obstacles in the domain are well captured. This confirms that CR3 performs also correctly in three dimensional heterogeneous media.

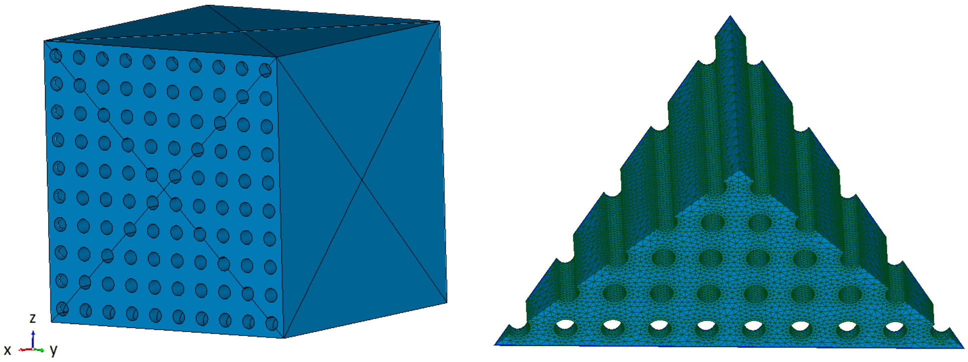


FIGURE 6.46: The three-dimensional reference cell (left) and one coarse element (right)

6.6.3 Applications to a periodic heterogeneous medium

We consider a flow past an array of periodically placed cylindrical obstacles in a parallelepiped $\Omega = [0,1] \times [0,1] \times [0,2]$. There are in total 5,000 horizontally placed cylinders in the domain. The diameter of cylinders is about $\varepsilon = 0.016$. We impose a parabolic velocity $\boldsymbol{u} = (0, 0, xy(1.0 - x)(1.0 - y))$ at the inlet and $\nabla \boldsymbol{u}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{0}$ at the outlet. The no-slip condition is imposed on other boundaries.

The medium is formed by an array of the reference cell shown in Figure 6.46 (A). The reference cell is a cubic domain containing 10×10 horizontally placed cylinders parallel to the x-axis. Making use of the periodicity of the medium, we use Algorithm 5.3 in Chapter 5 to generate necessary meshes and to compute multiscale basis functions. We need to solve local problems only on the reference cell shown in Figure 6.46 (A) and then copy the multiscale basis functions to other fine meshes. The reference cell is split into 24 tetrahedrons and one tetrahedron is shown in Figure 6.46 (B). Thus we need to solve local problems only on 24 coarse elements. In three dimensions, fine meshes on coarse elements must be fine enough in order to well resolve densely placed obstacles.

The coarse mesh consists of 6,000 tetrahedral coarse elements in total and each coarse element contains in average 100,000 tetrahedral small elements. There are about 6×10^8 tetrahedral small elements in the entire domain Ω. When using classical finite element, it is too expensive in terms of computing time to solve flow problems in such media.

TABLE 6.10: Computing time of different operations in Algorithm 5.3

Operation	Computing time
Solution of local problems	9 hours
Generation and translation of MED files	5 minutes
Copy of multiscale basis functions	80 minutes
Computation of matrices	10 minutes
Solution of the coarse-scale problem	1 minutes
Reconstruction of fine-scale solutions	10 minutes
Total computing time	$\sim$ 12 hours

6.6.3.1 Computing time

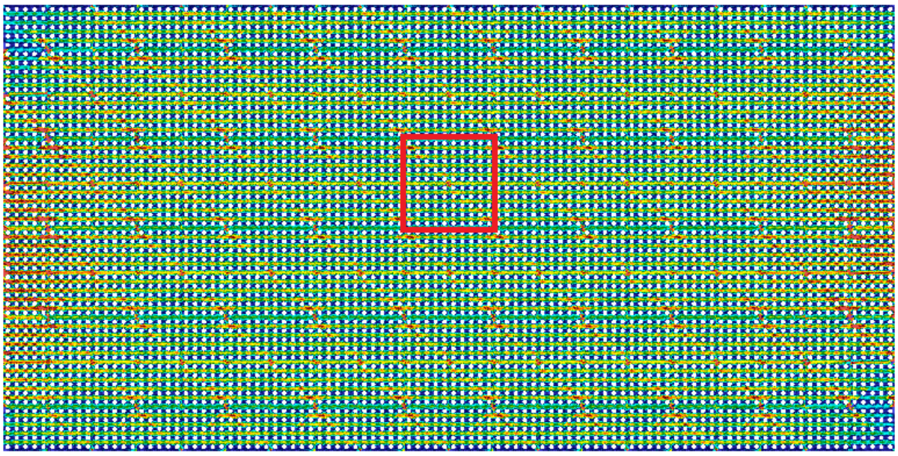
The entire numerical simulation using the multiscale simulation chain *SALOME-TrioCFD-VisIt* takes about 12 hours in total. We recall that we use Algorithm 5.3 to generate meshes and to compute multiscale basis functions in periodic media. The computing time consumed by each operation is presented in Table 6.10.

We remark that the most time-consuming operation is the solution of local problems which takes about 83% of the total computing time. We recall that these local problems are solved by the *prediction-correction* algorithm. The optimal performance of this algorithm is when each processor treats between 20,000 and 30,000 small elements. However, the number of small element in each coarse element exceeds largely 30,000. This explains why the solution of local problems is so time-consuming. To reduce the computing time, we can either develop a direct solver in three dimensions or develop the intra-cellular parallelism (see section 5.1) for the solution of local problems. At the end of this thesis, we implemented a direct solver only for two dimensions but not for three dimensions due to lack of time.

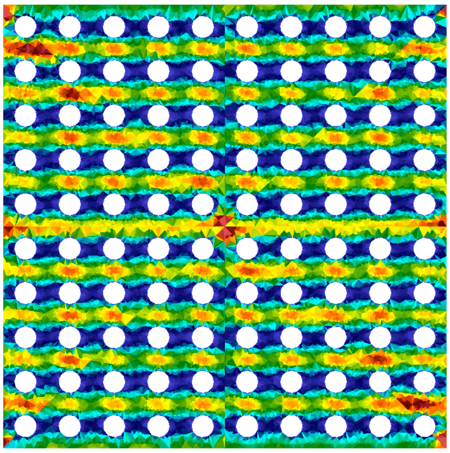
6.6.3.2 Storage of files

In multiscale simulations, we store mesh files (fine meshes and the coarse mesh), multiscale basis functions, matrices and reconstructed fine-scale solutions on the local disk. In particular, mesh files and multiscale basis functions need to be stored for reuse when boundary conditions or source terms of the coarse-scale problems change. These files take a total storage space of $400\,\mathrm{Go}$ on a local disk of $450\,\mathrm{Go}$, in which mesh files take about $50\,\mathrm{Go}$, basis functions and fine-scale solutions together take about $350\,\mathrm{Go}$.

One important factor that limits the application of multiscale finite element methods is the storage of files. The storage of a large number of files can exceed easily the limit of the local disk. It is important to note that the reconstructed fine-scale solutions are stored in the LATA format which takes a large storage space. Thus in practice, it is possible that we can only reconstruct fine-scale solutions in the regions of interest.



(A) Slice of velocity field $|\boldsymbol{u}|$ at $x = 0.5$



(B) A zoom

FIGURE 6.47: The slice of velocity field $|\boldsymbol{u}|$ at the plane $x = 0.5$

6.6.3.3 Numerical results

Figure 6.47 shows the slice of the velocity field and the zoom of a region at the plane $x = 0.5$. It is shown that important flow features are well captured by CR3 and the numerical result is rather reasonable. This underlines the good performance of multiscale methods in three-dimensional highly heterogeneous media.

Conclusions

In this chapter, we summarize the main results of this thesis from theoretical aspects to practical implementations of Crouzeix-Raviart multiscale finite element methods. Section 7.1 reviews some Crouzeix-Raviart multiscale finite element methods presented in this thesis and compare their accuracy. Section 7.2 reviews the *SALOME-TrioCFD-VisIt* multiscale simulation chain, including the parallelisms in the simulation chain, algorithms for the generation of meshes in *SALOME* and practical implementations in *TrioCFD*. Throughout this chapter, we review some remarks made during the thesis and propose some ideas and perspectives to improve the multiscale simulation chain.

7.1 Theoretical aspects

7.1.1 The Crouzeix-Raviart multiscale method defined by Stokes equations

We first review the Crouzeix-Raviart multiscale finite element method (see section 3.2) where multiscale basis functions are solutions of Stokes local problems. This method was originally proposed by [95, 118] and reviewed in detail in this thesis. Let d the dimension of the computational domain and s be a positive integer. Let $\boldsymbol{\omega}_{E,i} : E \to \mathbb{R}^d$ be some vector-valued functions associated to $E \in \mathcal{E}_H$ and $i = 1, \cdots, s$.

The Stokes local problems are associated to faces of the coarse mesh. For any face $E \in \mathcal{E}_H$ and $i = 1, \cdots, s$, we construct $\boldsymbol{\Phi}_{E,i} : \Omega^\varepsilon \to \mathbb{R}^d$ and the accompanying pressure $\pi_{E,i} : \Omega^\varepsilon \to \mathbb{R}$ such that $\boldsymbol{\Phi}_{E,i}$ and $\pi_{E,i}$ vanish outside the two coarse elements T_k which share the face E, for $k = \{1, 2\}$ (only one coarse element if $E \in \partial\Omega$) and solve on T_k:

$$\begin{cases} -\mu\Delta\boldsymbol{\Phi}_{E,i} + \nabla\pi_{E,i} = \mathbf{0} \text{ in } T_k \cap \Omega^\varepsilon, \\ \text{div } \boldsymbol{\Phi}_{E,i} = \alpha_{E,i} \text{ in } T_k \cap \Omega^\varepsilon, \\ \mu\nabla\boldsymbol{\Phi}_{E,i}\boldsymbol{n} - \pi_{E,i}\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\right\} \text{ on } F \cap \Omega^\varepsilon,\ \forall F \in \mathcal{E}(T_k), \\ \boldsymbol{\Phi}_{E,i} = \mathbf{0} \text{ on } \partial B^\varepsilon \cap T_k, \\ \int_{F\cap\Omega^\varepsilon} \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{\omega}_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases} \quad \forall F \in \mathcal{E}\left(T_k\right),\ j = 1, \cdots, s. \\ \int_{T_k\cap\Omega^\varepsilon} \pi_{E,i} = 0. \end{cases} \tag{7.1}$$

where $\mathcal{E}(T_k)$ is the set of faces of T_k. The constant $\alpha_{E,i}$ depends on T_k and satisfies the compatibility relation $\int_{T_k\cap\Omega^\varepsilon} \alpha_{E,i} = \int_{\partial(T_k\cap\Omega^\varepsilon)} \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{n}$.

In this thesis, weighting functions are chosen as: for any $E \in \mathcal{E}_H$,

$$\text{for } d = 2 : \begin{cases} s = 2 : \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2. \\ s = 3 : \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2,\ \boldsymbol{\omega}_{E,3} = \boldsymbol{n}_E\psi_E. \\ s = 4 : \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2,\ \boldsymbol{\omega}_{E,3} = \boldsymbol{n}_E\psi_E,\ \boldsymbol{\omega}_{E,4} = \boldsymbol{\tau}_E\phi_E. \end{cases} \tag{7.2}$$

$$\text{for } d=3: \begin{cases} s=3: \ \boldsymbol{\omega}_{E,1}=\boldsymbol{e}_1, \ \boldsymbol{\omega}_{E,2}=\boldsymbol{e}_2, \ \boldsymbol{\omega}_{E,3}=\boldsymbol{e}_3. \\ s=4: \ \boldsymbol{\omega}_{E,1}=\boldsymbol{e}_1, \ \boldsymbol{\omega}_{E,2}=\boldsymbol{e}_2, \ \boldsymbol{\omega}_{E,3}=\boldsymbol{e}_3, \ \boldsymbol{\omega}_{E,4}=\boldsymbol{n}_E\psi_E. \end{cases} \tag{7.3}$$

where $\{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_d\}$ is the canonical basis of $\mathbb{R}^d$ and $\boldsymbol{n}_E$ and $\boldsymbol{\tau}_E$ are respectively the unit vector normal and tangent to the face E. Besides, ψ_E and ϕ_E are linear polynomials which satisfy $\int_{E\cap\Omega^\varepsilon}\psi_E=0$ and $\int_{E\cap\Omega^\varepsilon}\phi_E=0$.

Consequently, the approximation space of velocity is formed by solutions of Stokes local problems defined by (7.1) and the approximation space of pressure contains element-wise constant pressure. In the original work [95, 118], the authors have used weighting functions in the cases of $s=2$ and $s=3$ defined by (7.2) and solved two-dimensional Stokes problems on the coarse mesh. In this thesis, in two dimensions, we try all the weighting functions defined by (7.2) and compare the accuracy of numerical results. Then we extend the multiscale method in three dimensions with two choices of weighting functions defined by (7.3).

Besides, we propose to solve not only Stokes problems, but also Oseen problems and Navier-Stokes problems by this method on the coarse mesh. We adapt the *Streamline-Upwind/Petrov-Galerkin* (SUPG) stabilization technique (see subsection 3.2.4) to the multiscale context in order to preclude oscillations and stabilize the solution of coarse-scale problems. The stabilization technique is necessary when solving Oseen or Navier-Stokes problems on the coarse mesh.

7.1.2 The Crouzeix-Raviart multiscale method defined by Oseen equations

Then we present a Crouzeix-Raviart multiscale finite element method (see section 3.3) where multiscale basis functions are solutions of Oseen local problems. This method was first proposed by [117] and some numerical experiments for small Oseen velocities were presented. However, no theoretical analysis was provided to show the well-posedness of local problems. In this thesis, we first tried to solve local problems proposed in [117] but numerical oscillations appeared in the solution for large Oseen velocities, even when using the Upwind scheme for the Oseen term. Thus we propose a new definition of local problems and prove their well-posedness. This improvement precludes oscillations in the solution of local problems for whatever Oseen velocities.

The Oseen local problems are associated to faces of the coarse mesh. For any $E\in\mathcal{E}_H$ and $i=1,\cdots,s$, we construct $\boldsymbol{\Phi}_{E,i}:\Omega^\varepsilon\to\mathbb{R}^d$ and the accompanying pressure $\pi_{E,i}:\Omega^\varepsilon\to\mathbb{R}$ such that $\boldsymbol{\Phi}_{E,i}$ and $\pi_{E,i}$ vanish outside the two coarse elements T_k which share the face E, $k\in\{1,2\}$ (only one coarse element if $E\in\partial\Omega$) and solve on T_k:

$$\begin{cases} -\mu\Delta\boldsymbol{\Phi}_{E,i}+\rho\left(\boldsymbol{U}_o\cdot\nabla\right)\boldsymbol{\Phi}_{E,i}+\nabla\pi_{E,i}=\boldsymbol{0} \text{ in } T_k\cap\Omega^\varepsilon, \\ \operatorname{div}\boldsymbol{\Phi}_{E,i}=\alpha_{E,i} \text{ in } T_k\cap\Omega^\varepsilon, \\ \mu\nabla\boldsymbol{\Phi}_{E,i}\boldsymbol{n}-\frac{1}{2}\rho\left(\boldsymbol{U}_o\cdot\boldsymbol{n}\right)\boldsymbol{\Phi}_{E,i}-\pi_{E,i}\boldsymbol{n}\in\operatorname{span}\left\{\boldsymbol{\omega}_{F,1},\cdots,\boldsymbol{\omega}_{F,s}\right\} \text{ on } F\cap\Omega^\varepsilon, \ \forall F\in\mathcal{E}(T_k), \\ \boldsymbol{\Phi}_{E,i}=\boldsymbol{0} \text{ on } \partial B^\varepsilon\cap T_k, \\ \int_{F\cap\Omega^\varepsilon}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{\omega}_{F,j}=\begin{cases}\delta_{ij}, & F=E \\ 0, & F\neq E\end{cases} \quad \forall F\in\mathcal{E}\left(T_k\right), \ j=1,\cdots,s. \\ \int_{T_k\cap\Omega^\varepsilon}\pi_{E,i}=0. \end{cases} \tag{7.4}$$

where $\mathcal{E}(T_k)$ is the set of faces of T_k. The constant $\alpha_{E,i}$ depends on T_k and satisfies $\int_{T_k\cap\Omega^\varepsilon}\alpha_{E,i}=\int_{\partial(T_k\cap\Omega^\varepsilon)}\boldsymbol{\Phi}_{E,i}\cdot\boldsymbol{n}$.

Consequently, the approximation space of velocity is formed by solutions of Oseen local problems defined by (7.4) and the approximation space of pressure contains element-wise constant pressure. In the work [117], the author tried only weighting functions in the case of $s=2$ defined by (7.2) and solved only Oseen problems on the coarse mesh. In this thesis, we try all the weighting functions defined by (7.2) and compare the accuracy of numerical results. Besides, we propose to solve not only Oseen problems, but also Navier-Stokes problems by this method on the coarse mesh. We adapt the SUPG stabilization technique (see subsection 3.3.5) to the multiscale context in order to preclude oscillations and stabilize solutions of Navier-Stokes problems on the coarse mesh. We remark that the stabilization technique is not necessary when solving Oseen problems on the coarse mesh since no oscillations appeared in the solution.

7.1.3 The Crouzeix-Raviart multiscale method defined by both Stokes and Oseen local solutions

In order to improve the accuracy of the Crouzeix-Raviart multiscale method where multiscale basis functions are either defined by Stokes (7.1) or Oseen equations (7.4), we propose a Crouzeix-Raviart multiscale finite element method (see section 3.4) where the approximation space of velocity is formed by the union of basis functions defined by Stokes and Oseen equations. The approximation space of pressure still contains element-wise constant pressure.

In numerical simulations, we solve Stokes problems and Oseen problems on the coarse mesh with this multiscale method. Contrary to expectations, our numerical results revealed that the Crouzeix-Raviart multiscale method where basis functions are defined by both Stokes and Oseen equations is more accurate than the multiscale method defined by Stokes equations alone but much less accurate than that defined by Oseen equations alone. The reasons for this result is not entirely understood in this thesis.

7.1.4 The Crouzeix-Raviart multiscale method enriched by bubble functions

In order to improve the accuracy of the Crouzeix-Raviart multiscale method where multiscale basis functions are solutions of Stokes local problems defined by (7.1), we propose a Crouzeix-Raviart multiscale finite element method (see section 3.5) where the velocity approximation space is enriched by adding bubble functions. Inspired by bubble functions proposed by [54, 102, 105, 113] for diffusion or diffusion-advection equations, we propose for the first time to define bubble functions for Stokes equations.

The bubble functions are associated to elements of the coarse mesh. For each coarse element $T \in \mathcal{T}_H$ and for $k = 1, \cdots, r$, the support of $\boldsymbol{\Psi}_{T,k}$ is reduced to $T \cap \Omega^\varepsilon$. We construct $\boldsymbol{\Psi}_{T,k} : \Omega^\varepsilon \to \mathbb{R}^d$ and $\pi_{T,k} : \Omega^\varepsilon \to \mathbb{R}$ as solutions of

$$\begin{cases} -\mu\Delta\boldsymbol{\Psi}_{T,k} + \nabla\pi_{T,k} = \boldsymbol{e}_k \text{ in } T \cap \Omega^\varepsilon, \\ \text{div } \boldsymbol{\Psi}_{T,k} = \alpha_{T,i} \text{ in } T \cap \Omega^\varepsilon, \\ \mu\nabla\boldsymbol{\Psi}_{T,k}\boldsymbol{n} - \pi_{T,k}\boldsymbol{n} \in \operatorname{span}\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\} \text{ on } F \cap \Omega^\varepsilon,\ \forall F \in \mathcal{E}(T), \\ \boldsymbol{\Psi}_{T,k} = \boldsymbol{0} \text{ on } \partial B^\varepsilon \cap T, \\ \int_{F\cap\Omega^\varepsilon} \boldsymbol{\Psi}_{T,k} = \boldsymbol{0}\ \forall F \in \mathcal{E}\left(T\right), \\ \int_{T\cap\Omega^\varepsilon} \pi_{T,k} = 0. \end{cases} \tag{7.5}$$

where $\mathcal{E}(T)$ is the set of faces composing ∂T and the constant $\alpha_{T,k}$ satisfies $\int_{T\cap\Omega^\varepsilon} \alpha_{T,k} = \int_{\partial(T\cap\Omega^\varepsilon)} \boldsymbol{\Psi}_{T,k} \cdot \boldsymbol{n}$.

In two-dimensional numerical experiments performed in this thesis, we choose the following weighting functions:

$$\text{for } d = 2: \begin{cases} s = 2: \ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2. \\ r = 2: \ \boldsymbol{\varphi}_{T,1} = \boldsymbol{e}_1,\ \boldsymbol{\varphi}_{T,2} = \boldsymbol{e}_2. \end{cases} \tag{7.6}$$

Weighting functions in the case of $s = 2$ are used in Stokes local problems (7.1) and those in the case of $r = 2$ are used in the computation of bubble functions defined by (7.5).

Consequently, the approximation space of velocity is formed by solutions of Stokes local problems defined by (7.1) and bubble functions defined by (7.5). The approximation space of pressure contains element-wise constant pressure. However, our numerical experiments show that the addition of bubble functions does not improve the accuracy of multiscale methods. Thus we perform serious theoretical analysis to verify the construction of the multiscale approximation spaces of velocity and pressure. Our theoretical analysis show that the approximation spaces of velocity and pressure need to be redefined in order to effectively improve the accuracy of the method. In particular, the approximation space of pressure needs to be enriched by some weighting functions which are defined by polynomials. Thus we propose an original high-order Crouzeix-Raviart multiscale method which is presented in what follows.

7.1.5 The high-order Crouzeix-Raviart multiscale finite element method

We propose an original Crouzeix-Raviart multiscale finite element method (see section 3.6) to provide a general framework which defines proper approximation spaces of velocity and pressure for higher accuracy. As far as we know, this is the first time that the high-order Crouzeix-Raviart multiscale finite element method is proposed for flow problems.

For any integer n and any integer $1 \leq l \leq d$, we denote by $\mathbb{P}_n^l$ the linear space spanned by l-variate polynomial functions of degree at most n. The dimension of $\mathbb{P}_n^l$ is

$$N_n^l := \dim\left(\mathbb{P}_n^l\right) = \binom{n+l}{n}$$

For any $T \in \mathcal{T}_H$, we denote by $\mathbb{P}_n^d\,(T)$ the restriction to T of polynomials in $\mathbb{P}_n^l$. For any $F \in \mathcal{E}_H$, we denote by $\mathbb{P}_n^{d-1}\,(F)$ the restriction to F of polynomials of $\mathbb{P}_n^l$. For the sake of simplicity, we denote $\mathbb{P}_n^d\,(T)$ and $\mathbb{P}_n^{d-1}\,(F)$ respectively by $\mathbb{P}_n\,(T)$ and $\mathbb{P}_n\,(F)$.

We define weighting functions for the velocity and pressure in what follows. Let s be a positive integer and d be the dimension of heterogeneous media. We associate the vector-valued function $\boldsymbol{\omega}_{E,i}$: $E \rightarrow \mathbb{R}^d$ to each face $E \in \mathcal{E}_H$ and $i = 1, \cdots, s$. Let r be a positive integer. We associate the vector-valued functions $\boldsymbol{\varphi}_{T,k} : T \rightarrow \mathbb{R}^d$ to each coarse element $T \in \mathcal{T}_H$ and $k = 1, \cdots, r$. Let t be a positive integer. We associate the scalar functions $\varpi_{T,j}$: $T \rightarrow \mathbb{R}$ to each coarse element $T \in \mathcal{T}_H$ and $j = 1, \cdots, t$. These weighting functions are defined by polynomials of arbitrary orders n as shown in Assumption 3.6.1 and Assumption 3.6.2.

Local problems in high-order Crouzeix-Raviart multiscale methods are associate to both faces and elements of the coarse mesh. We first construct the basis function associated to faces of the coarse mesh. For any $E \in \mathcal{E}_H$ and for $i = 1, \cdots, s$, find the function $\boldsymbol{\Phi}_{E,i} : \Omega^\varepsilon \rightarrow \mathbb{R}^d$, the pressure $\pi_{E,i} : \Omega^\varepsilon \rightarrow \mathbb{R}$ such that $\boldsymbol{\Phi}_{E,i}$ and $\pi_{E,i}$ vanish outside the two coarse elements T_k which share the face E for $k \in \{1, 2\}$ (only one coarse element if $E \in \partial\Omega$) and solve on T_k:

$$\begin{cases} -\mu\Delta\boldsymbol{\Phi}_{E,i} + \nabla\pi_{E,i} \in \operatorname{span}\left\{\boldsymbol{\varphi}_{T_k,1}, \cdots, \boldsymbol{\varphi}_{T_k,r}\right\} \text{ in } T_k \cap \Omega^\varepsilon \\ \operatorname{div} \boldsymbol{\Phi}_{E,i} \in \operatorname{span}\left\{\varpi_{T_k,1}, \cdots, \varpi_{T_k,t}\right\} \text{ in } T_k \cap \Omega^\varepsilon \\ \mu\nabla\boldsymbol{\Phi}_{E,i}\boldsymbol{n} - \pi_{E,i}\boldsymbol{n} \in \operatorname{span}\left\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\right\} \text{ on } F,\ \forall F \in \mathcal{E}(T_k) \\ \boldsymbol{\Phi}_{E,i} = \mathbf{0} \text{ on } \partial B^\varepsilon \cap T_k \\ \int_{F\cap\Omega^\varepsilon} \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{\omega}_{F,j} = \begin{cases} \delta_{ij}, & F = E \\ 0, & F \neq E \end{cases} \quad \forall F \in \mathcal{E}\left(T_k\right),\ \forall j = 1, \cdots, s \\ \int_{T_k\cap\Omega^\varepsilon} \boldsymbol{\Phi}_{E,i} \cdot \boldsymbol{\varphi}_{T_k,l} = 0 \ \forall l = 1, \cdots, r \\ \int_{T_k\cap\Omega^\varepsilon} \pi_{E,i} \cdot \varpi_{T_k,m} = 0 \ \forall m = 1, \cdots, t \end{cases} \tag{7.7}$$

Now we construct the basis function associated to elements of the coarse mesh. For each $T \in \mathcal{T}_H$ and for $k = 1, \cdots, r$, the support of the function $\boldsymbol{\Psi}_{T,k}$ is reduced to $T \cap \Omega^\varepsilon$. We find

$\mathbf{\Psi}_{T,k} : \Omega^\varepsilon \to \mathbb{R}^d$ and $\pi_{T,k} : \Omega^\varepsilon \to \mathbb{R}$ by solving on T:

$$\begin{cases} -\mu\Delta\mathbf{\Psi}_{T,k} + \nabla\pi_{T,k} \in \text{span}\left\{\boldsymbol{\varphi}_{T,1}, \cdots, \boldsymbol{\varphi}_{T,r}\right\} \text{ in } T \cap \Omega^\varepsilon \\ \text{div } \mathbf{\Psi}_{T,k} \in \text{span}\left\{\varpi_{T,1}, \cdots, \varpi_{T,t}\right\} \text{ in } T \cap \Omega^\varepsilon \\ \mu\nabla\mathbf{\Psi}_{T,k}\boldsymbol{n} - \pi_{T,k}\boldsymbol{n} \in \text{span}\left\{\boldsymbol{\omega}_{F,1}, \cdots, \boldsymbol{\omega}_{F,s}\right\} \text{ on } F,\ \forall F \in \mathcal{E}(T) \\ \mathbf{\Psi}_{T,k} = 0 \text{ on } \partial B^\varepsilon \cap T \\ \int_{F\cap\Omega^\varepsilon} \mathbf{\Psi}_{T,k} \cdot \boldsymbol{\omega}_{F,j} = 0\ \forall F \in \mathcal{E}\left(T\right),\ \forall j = 1, \cdots, s \\ \int_{T\cap\Omega^\varepsilon} \mathbf{\Psi}_{T,k} \cdot \boldsymbol{\varphi}_{T,l} = \delta_{kl}\ \forall l = 1, \cdots, r \\ \int_{T\cap\Omega^\varepsilon} \pi_{E,i} \cdot \varpi_{T,m} = 0\ \forall m = 1, \cdots, t \end{cases} \tag{7.8}$$

In this thesis, we have performed two-dimensional numerical simulations with weighting functions below:

$$\text{for } n = 1,\ d = 2:\ \begin{cases} s = 4:\ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2,\ \boldsymbol{\omega}_{E,3} = \boldsymbol{n}_E\psi_E,\ \boldsymbol{\omega}_{E,4} = \boldsymbol{\tau}_E\phi_E. \\ t = 3:\ \varpi_{T,1} = 1,\ \varpi_{T,2} = x,\ \varpi_{T,3} = y. \\ r = 2:\ \boldsymbol{\varphi}_{T,1} = \boldsymbol{e}_1,\ \boldsymbol{\varphi}_{T,2} = \boldsymbol{e}_2. \end{cases}$$

where $\{\boldsymbol{e}_1,\ \boldsymbol{e}_2\}$ is the canonical basis of $\mathbb{R}^2$ and $\boldsymbol{n}_E$ and $\boldsymbol{\tau}_E$ are respectively the unit vector normal and tangent to the face E. Besides, $\psi_E \in \mathbb{P}_1(E)$ and $\phi_E \in \mathbb{P}_1(E)$ which satisfy that $\int_{E\cap\Omega^\varepsilon} \psi_E = 0$ and $\int_{E\cap\Omega^\varepsilon} \phi_E = 0$.

For $n = 0$, for any $T \in \mathcal{T}_H$, weighting functions are chosen as:

$$\text{for } d = 2,\ n = 0:\ \begin{cases} s = 2:\ \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1,\ \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2. \\ t = 1:\ \varpi_{T,1} = 1 \\ r = 1:\ \boldsymbol{\varphi}_{T,1} = 0 \end{cases}$$

In this case, we remark that the high-order multiscale method is exactly the multiscale method defined by (7.1) with weighting functions in the case of $s = 2$ defined by (7.2).

Consequently, the approximation space of velocity is formed by solutions of local problems defined by (7.7)–(7.8). The approximation space of pressure contains pressure in the $\mathbb{P}_n(T)$ space instead of $\mathbb{P}_0(T)$ for any $T \in \mathcal{T}_H$. By increasing the value of n, we construct more basis functions by solving system (7.7)–(7.8) and thus both the approximation space of velocity and pressure will be further enriched. Our numerical simulations confirm that high-order multiscale finite element methods can improve significantly the accuracy of both the velocity and pressure.

In conclusion, the high-order multiscale method provides a more general definition of approximation spaces than that proposed by [95, 117, 118]. By varying the degrees of polynomials in the definition of weighting functions, this multiscale method permits to find a compromise between the desired accuracy and the computational costs. Similarly, we define also the high-order multiscale finite element method based on Oseen equations (see section 3.7).

In this thesis, only the high-order multiscale method for $n = 1$ is implemented in *TrioCFD* and validated by numerical experiments. In the future work, an error analysis should be made for the high-order multiscale method in order to clarify the relation of errors with the degree of polynomials. It would be interesting to implement the high-order method for $n = 2$, $n = 3$ and so on and compare its performance with other CR multiscale methods.

7.1.6 Comparison of Crouzeix-Raviart multiscale finite element methods

We performed some numerical experiments to study the numerical convergence of Crouzeix-Raviart multiscale methods presented in this thesis. We fix the element size h of fine meshes, the size obstacles ε and vary only the element size H of the coarse mesh. We ensure that these three sizes verify that $h \ll \varepsilon < H$. The element size h verifies $h \ll \varepsilon$ in order to capture the presence of small obstacles in the media. Besides, we are interested only in the case $H > \varepsilon$ since the opposite case $H < \varepsilon$ is covered by classical finite element methods. Numerical convergence studies (see subsection 6.4.1 and subsection 6.4.2) in non-periodic two-dimensional heterogeneous media show good convergence of Crouzeix-Raviart multiscale methods toward reference solutions. It is shown that Crouzeix-Raviart multiscale methods can well capture the presence of small obstacles and important macroscopic flow features in heterogeneous media on rather coarse mesh.

As expected, high-order multiscale methods produce the most accurate velocity and pressure. Besides, we have compared solutions given by high-order Crouzeix-Raviart multiscale methods where local problems are discretized by Crouzeix-Raviart element or $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ element. It is shown that the fine-scale velocity is the same but interestingly, the fine-scale pressure is much more accurate when local problems are discretized by the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ element. In high-order multiscale methods, when increasing degrees of polynomials in weighting functions (see subsection 3.6.1), it would be important to use higher-order finite elements in local problems.

Numerical experiments (subsection 6.4.1.2) show that the multiscale method enriched by bubble functions does not improve the accuracy of numerical results. This result is consistent with theoretical analysis presented in section 3.5 and it further strengthens our conviction that high-order multiscale methods provide the correct definition of bubble functions for Stokes or Oseen equations.

As presented in subsection 6.4.2.3, when solving Oseen problems on the coarse mesh, contrary to expectations, the multiscale method defined by both Stokes and Oseen equations is more accurate than that defined by Stokes equations (7.1) alone but much less accurate than that defined by Oseen equations (7.4) alone. The reasons for this result are not yet completely understood.

Numerical experiments (see subsection 6.4.2.2 and subsection 6.4.3) show that the multiscale method defined by Oseen equations (7.4) with weighting functions in the case of $s = 2$ or $s = 3$ defined by (7.2), and the multiscale method defined by Stokes equations (7.4) with weighting functions in the case of $s = 3$ defined by (7.2) are both rather accurate for solving Navier-Stokes problems or Oseen problems on the coarse mesh. With weighting functions in the case of $s = 2$ defined by (7.2), the multiscale method defined by Stokes equations is sensitive to Reynolds numbers and becomes much less accurate than the multiscale method defined by Oseen equations. We are aware that in our numerical experiments, fine-scale Reynolds numbers Re^{ε} defined by subsection 6.4.3 are not very large and we focus only on steady-state Navier-Stokes problems on the coarse mesh. In the future work, it would be very interesting to extend the multiscale method to solve unsteady Navier-Stokes problems with larger Reynolds numbers.

In two-dimensional periodic heterogeneous media, numerical experiments (see section 6.5) reveal that the multiscale method defined by Stokes equations with weighting functions in the case of $s = 2$ defined by (7.2) does not produce correct solutions. However, weighting functions in the case of $s = 3$ and $s = 4$ can both improve significantly the accuracy of numerical results. We conclude that it is necessary to use weighting functions in the case of $s = 3$ or $s = 4$ in periodic heterogeneous media in order to obtain accurate numerical results.

Two- and three-dimensional demonstrative applications shown in subsection 6.5.3 and section 6.6 confirm that multiscale methods can capture successfully fine-scale flow features and the presence of small obstacles in highly heterogeneous media. We are aware that three-dimensional heterogeneous media remain relatively simple compared to the structures of nuclear fuel assemblies which are extremely complicated. In the future work, an important work could be devoted the development of algorithms in order to generate meshes required by MsFEMs on such complicated structures.

7.2 The multiscale simulation chain *SALOME-TrioCFD-VisIt*

The multiscale simulation chain consists of all necessary tools and preliminary preparations for numerical simulations with multiscale finite element methods. *SALOME* is used in the first step of the simulation chain to prepare necessary mesh files. *TrioCFD* is involved in the second step to perform numerical simulations with Crouzeix-Raviart multiscale methods. *VisIt* is used to visualize the reconstructed fine-scale solutions.

7.2.1 The intra- and extra-cellular parallelisms

In the multiscale simulation chain, as shown in Figure 5.1, we distinguish two types of parallelisms: the intra-cellular parallelism and the extra-cellular parallelism. Both parallelisms contribute to the reduction of computing time and memory. The extra-cellular parallelism is an intrinsic parallelism of multiscale finite element methods. As presented in Chapter 3, Crouzeix-Raviart multiscale finite element methods are nonconforming methods, i.e. only the average jump of velocity is required to be continuous on the interface between coarse elements. Thus local problems are solved on each coarse element independently from adjacent ones. Similarly, the computation of matrices and the reconstruction of fine-scale solutions are carried out locally on each coarse element. we developed efficient algorithms in *SALOME* which allows to generate fine meshes rapidly on one coarse element after another. Therefore the generation of fine meshes does not benefit from the extra-cellular parallelism.

SALOME have parallized meshing tools which allow to assign several processors for one task. For very complicated geometries, fine meshes can be of large size and it would be necessary to parallize algorithms developed in this thesis to benefit from the intra-cellular parallelism. In numerical simulations carried out in this thesis, fine meshes are not of very large size and can be generated rapidly using even only one processor in *SALOME*.

Local problems are solved in *TrioCFD* by adapting the *prediction-correction* algorithm (see section 4.3) or by the direct solver (see section 4.6) implemented in this thesis. In numerical experiments, we remark that the solution of three-dimensional large systems is time-consuming when using the *prediction-correction* algorithm. Similarly, it is possible that coarse-scale problems might be of large size in very complicated applications. Thus in the future, it is interesting to develop the intra-cellular parallelism to solve local and coarse-scale problems and reduce the computing time. The computation of matrices and the reconstruction of fine-scale solutions are carried out sequentially on coarse elements and the intra-cellular parallelism is not necessary.

7.2.2 Generation of meshes in *SALOME*

In this thesis, we developed three algorithms to generate necessary meshes for Crouzeix-Raviart multiscale methods. Shown in Algorithm 5.1, the first algorithm creates conforming fine meshes where mesh nodes match on the interface between coarse elements. The union of conforming fine meshes forms the fine reference mesh which is thus a conforming mesh. For the validation of Crouzeix-Raviart multiscale methods, we compare numerical solutions computed by multiscale methods to those computed by the classical finite volume element method on the reference mesh.

As is well known, the reference mesh is not required in multiscale methods and is needed only for validation purpose. Shown in Algorithm 5.2, the second algorithm is devoted to the generation of fine meshes where mesh nodes do not necessarily match on the interface between coarse elements. We generate fine meshes on one coarse element independently from adjacent ones.

Shown in Algorithm 5.3, the third algorithm is specially developed for periodic heterogeneous media. This algorithm facilitates the generation of fine meshes and multiscale basis functions by taking advantage of the periodicity of the media. A periodic medium can be considered as a periodic array of the reference cell and multiscale basis functions computed in the reference cell are the same as in other cells. Thus we enerate fine meshes and compute multiscale basis functions only in the reference cell and then copy them to the entire domain.

We remarked one difficulty for generating meshes required by Crouzeix-Raviart multiscale methods. As discussed in subsection 5.2.7, obstacles may be tangent to the interface between coarse elements, especially when obstacles are densely packed in heterogeneous media. In this case, meshing tools may fail to generate meshes due to isolated tangent points. We envision two solutions to this problem in the future. The first is to mesh obstacles and use the penalization technique to approximate the no-slip boundary conditions on obstacles. The second is to develop a coarse mesh consisting of polygons whose forms can change to avoid generating tangent points. Theoretically, in multiscale finite elements, elements of the coarse mesh can be of any shape.

7.2.3 Implementations in *TrioCFD*

In the work [95, 118], local problems are discretized by simple Cartesian meshes and the penalty technique is used to impose the no-slip boundary condition on the boundary of obstacles. In this thesis, Crouzeix-Raviart multiscale finite element methods are implemented in *TrioCFD* [139] which focus on the finite volume element method. We use boundary-fitted unstructured meshes made of triangular or tetrahedral elements. No penalization technique is used and the no-slip boundary condition is imposed directly on the boundary of obstacles.

Local problems are discretized by the finite volume element method and physical unknowns are discretized by the Crouzeix-Raviart finite element (see section 4.1). In particular, local problems in high-order multiscale methods are also discretized by the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ element (see subsection 4.6.1). Numerical simulations confirm that using $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ element, the pressure is more accurate than using the Crouzeix-Raviart element. In the future work, the high-order multiscale finite element methods with $n = 2$, $n = 3$ and so on should be implemented and tested to better understand the relation between the rate of convergence of the method and the degree of polynomials n. When n is larger, it would be necessary to implement higher-order finite elements in *TrioCFD* to discretize local problems.

The computation of matrices and the reconstruction of fine-scale solutions are not native functions in *TrioCFD*. As presnted in section 4.4, when computing matrices locally on coarse elements, we decompose multiscale basis functions using the Crouzeix-Raviart finite element or the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element basis functions. The properties of these finite element basis functions facilitate the computation of integrals in the coefficients of matrices. Coarse-scale problems are assembled and discretized by a Galerkin method where basis functions are solutions of local problems defined by, for example (7.1), (7.4), (7.7) and so on.

In an early stage of this thesis, the *prediction-correction* algorithm is modified to solve local and coarse-scale problems. This algorithm is a time-marching technique which solves steady problems through a pseudo-transient process which can be sometimes time-consuming. We remark that this algorithm is not suitable for solving local and coarse-scale problems in multiscale methods. Thus in a later stage of this thesis, an important work was devoted to the implementation of a direct solver in two dimensions. The extension of this solver in three dimensions is straightforward and could be carried out in the future.

Appendix A

Source files and data files in the multiscale simulation chain

We present in this appendix one simplified Python script which describes the generation of conforming meshes presented in subsection 5.2.4. We give a commented example of data files for solving local problems and coarse-scale problems in *TrioCFD*. We also provide a Python script for the visualization of fine-scale solutions in *VisIt*.

A.1 A Python script for generating meshes in *SALOME*

In Chapter 5, we have presented three algorithms to generate necessary meshes in *SALOME* for multiscale finite element methods. Here we present a simplified Python script which describes main steps of Algorithm 5.1. Using this script, we generate a coarse mesh, a reference mesh and conforming fine meshes in a three-dimensional heterogeneous medium with periodically placed cylinders.

```python
import os, salome, math
from salome.geom import geomBuilder
geompy = geomBuilder.New(salome.myStudy)
from salome.smesh import smeshBuilder
mesher = smeshBuilder.New(salome.myStudy)

# Define some global parameters
# ----------------------------------------
# define cylinders
origin_x = 0
origin_y = 0
radius = 0.003
pas_crayon = 0.0126
etf = pas_crayon - 2*radius

# define the unit cell
grid_x0  = 0.0
grid_y0  = 0.0
grid_z0  = 0.0
grid_nx = 3   # number of cylinders = grid_nx * grid_ny
grid_ny = 3
grid_x = origin_x+pas_crayon*(grid_nx-1)
grid_y = grid_x
grid_z = grid_x

# id_split = 3: cut the cell into 24 tetras
id_split = 3

# translation of unit cell
ts_nx = 0
ts_ny = 0
```

```
ts_nz = 4

# directory of the export of mesh files
grid_directory ='.'
python_file = os.path.join(grid_directory , "grid" + ".py")
med_files = os.path.join(grid_directory , "MED_files")
if not os.path.exists(med_files):
    os.makedirs(med_files , 0755)

# Build a tetrahedron from its vertices
# ------------------------------------------
tsl_x = "x"
tsl_z = "z"
def makeTetra(points):
  faces = []
    tetra_tsl = []
    for i in xrange(len(points)):
        sommets  = points[:i] + points[i+1:]
        triangle = geompy.MakePolyline(sommets, True)
        face     = geompy.MakeFace(triangle , True)
        faces.append(face)

    shell = geompy.MakeShell(faces)
    tetra = geompy.MakeSolid(shell)
    tetot.append(tetra)

  # translate tetrahedrons in X, Y, Z directions
    if ts_nx>0:
        for j in xrange(ts_nx):
            name_x = geompy.MakeTranslation(tetra ,(j+1)*grid_x ,0 ,0)
            tetot.append(name_x)
    if ts_ny>0:
        for m in xrange(len(tetra_tsl)):
            for k in xrange(ts_ny):
                name_y = tsl_x + str(m+1)
                name_y = name_y+str(k+1)
                name_y = geompy.MakeTranslation(tetot[m],0,(k+1)*grid_y ,0)
                tetot.append(name_y)
    if ts_nz>0:
        for p in xrange(len(tetra_tsl)):
            for q in xrange(ts_nz):
                name_z = tsl_z + str(p+1)
                name_z = name_z+str(q+1)
                name_z = geompy.MakeTranslation(tetot[p],0,0,(q+1)*grid_z)
                tetot.append(name_z)
    return tetot

# Make a group of solids or faces
# -----------------------------------
def buildGroup(geom, subshapes, name, solid_group):
    if solid_group:
        typ = geompy.ShapeType["SOLID"]
    else:
        typ = geompy.ShapeType["FACE"]
    group = geompy.CreateGroup(geom, typ)
    group.SetName(name)
    geompy.addToStudyInFather(geom, group, name)
    geompy.UnionList(group, subshapes)
    return group

# Build a group from a plane
# -----------------------------
def buildGroupShape(geom, name,  normal, x, y, z):
```

```python
    point      = geompy.MakeVertex(x, y, z)
    subshapes = geompy.GetShapesOnPlaneWithLocation(geom, geompy.ShapeType["EDGE"], normal, point, geomBuilder.GEOM.ST_ON)
    return buildGroup(geom, subshapes, name, False)

def buildGroupCylinders(geom, name, solides=False):
    if solides:
        typ = geompy.ShapeType["FACE"]
    else:
        typ = geompy.ShapeType["EDGE"]
    group = geompy.CreateGroup(geom, typ)
    group.SetName(name)
    geompy.addToStudyInFather(geom, group, name)
    return group

def addGroup(group, type_shape, kind, take):
    shapes = geompy.SubShapeAll(group.GetMainShape(), type_shape)
    for shape in shapes:
        is_kind = (geompy.KindOfShape(shape)[0] == kind )
        if ( take and is_kind ) or ((not take) and (not is_kind) ):
            geompy.UnionList(group, [shape])

# Build a geometric cube
# ----------------------
cube = geompy.MakeBox(0, 0, 0,  grid_x, grid_y, grid_z)
geompy.addToStudy(cube, "mesh_cube")

# Mesh the cube in one hexahedron
# -------------------------------
mesh_cube = mesher.Mesh(cube)
mesh_1d = mesh_cube.Segment()
mesh_1d.NumberOfSegments(1)
mesh_cube.Quadrangle()
mesh_cube.Hexahedron()
mesh_cube.Compute()

# Split the hexahedron in 6 or 24 tetrahedrons
# --------------------------------------------
mesh_cube.SplitVolumesIntoTetra(mesh_cube, id_split)

# Build geometric cylinders
# -------------------------
base = geompy.MakeVertex(origin_x, origin_y, 0)
axis = geompy.MakeVectorDXDYDZ(0, 0, 1)
list_cylinders = []
cyl0 = geompy.MakeCylinder(base, axis, radius, grid_z)
list_cylinders.append(cyl0)
cylinders = geompy.MakeCompound(list_cylinders)
geompy.addToStudy(cylinders, "cylinders")

# Convert the tetrahedrons of mesh_cube into geometric tetrahedrons
# -----------------------------------------------------------------
tetras = []
cells  = mesh_cube.GetElementsByType(smeshBuilder.VOLUME)
for cell in cells:
    nodes  = mesh_cube.GetElemNodes(cell)
    coords = [ mesh_cube.GetNodeXYZ(node) for node in nodes ]
    points = [ geompy.MakeVertex(x, y, z) for x, y, z in coords ]
    tetra_tsl = makeTetra(points)
    tetras=tetras+tetra_tsl

# Build the gros_geo and ref_geo
gros_geo = geompy.MakePartition(tetras, [])
```

```python
ref_geo = geompy.MakeCut(gros_geo, cylinders)

# create geometric groups on ref_geo
# ----------------------------------------
nx = geompy.MakeVectorDXDYDZ(1, 0, 0)
ny = geompy.MakeVectorDXDYDZ(0, 1, 0)
nz = geompy.MakeVectorDXDYDZ(0, 0, 1)
inlet  = buildGroupShape(ref_geo,"inlet",nz,0,0,(ts_nz+1)*grid_z)
outlet = buildGroupShape(ref_geo,"outlet",nz,0,0,0)
up     = buildGroupShape(ref_geo,"up",ny,0,(ts_ny+1)*grid_y+etf+radius,0)
down   = buildGroupShape(ref_geo,"down",ny,0,-(etf+radius),0)
left   = buildGroupShape(ref_geo,"left",nx,-(etf+radius),0,0)
right = buildGroupShape(ref_geo,"right",nx,(ts_nx+1)*grid_x+etf+radius,0,0)

cylinders2d = buildGroupCylinders(reference, "cylinders2d", False)
addGroup(cylinders2d, geompy.ShapeType["FACE"], geompy.kind.CYLINDER2D, True)

# create geometric groups on gros_geo
# ----------------------------------------
IN  = buildGroupShape(gros_geo,"inlet",nz,0,0,(ts_nz+1)*grid_z)
OUT = buildGroupShape(gros_geo,"outlet",nz,0,0,0)
UP  = buildGroupShape(gros_geo,"up",ny,0,(ts_ny+1)*grid_y+etf+radius,0)
DOWN = buildGroupShape(gros_geo,"down",ny,0,-(etf+radius),0)
LHS = buildGroupShape(gros_geo,"left",nx,-(etf+radius),0,0)
RHS = buildGroupShape(gros_geo,"right",nx,(ts_nx+1)*grid_x+etf+radius,0,0)

# Build triangular faces
# --------------------------
ind = 0
triangles = { }
for face in geompy.SubShapeAll(gros_geo, geompy.ShapeType["FACE"]):
    compound = geompy.GetInPlace(ref_geo, face, True)
    for face1 in geompy.SubShapeAll(compound, geompy.ShapeType["FACE"]):
        face_id = geompy.GetSubShapeID(ref_geo, face1)
        triangles[face_id] = "Tri_"+str(ind)
    ind +=1

# Build 2D and 3D geometric groups
# ----------------------------------------
cylindersname = "Cylinders"
groups_2d = []
for face in geompy.SubShapeAll(ref_geo, geompy.ShapeType["FACE"]):
    if geompy.KindOfShape(face)[0] == geompy.kind.CYLINDER2D:
        name = cylindersname
    else:
        face_id = geompy.GetSubShapeID(ref_geo, face)
        name = triangles[face_id]
    group_2d = buildGroup(ref_geo, [face], name, False)
    groups_2d.append(group_2d)

groups_3d = []
i = 0
for solid in geompy.SubShapeAll(ref_geo, geompy.ShapeType["SOLID"]):
    group_3d = buildGroup(ref_geo, [solid], "tetra_"+str(i), True)
    groups_3d.append(group_3d)
    i += 1

# Generate the reference mesh
# ------------------------------
mesh_tet = mesher.Mesh(ref_geo)
algo_2d = mesh_tet.Triangle(algo=smeshBuilder.MG_CADSurf)
algo_2d.SetPhysicalMesh(1)
algo_2d.SetMaxSize(0.001)
```

```
algo_2d.SetPhySize(0.001)
algo_2d.SetGeometricMesh(1)
algo_2d.SetAngleMesh(5.0)
algo_2d.SetGradation(1.1)
algo_2d.SetVolumeGradation(True, 1.2)
for group_3d in groups_3d:
    algo_3d = mesh_tet.Tetrahedron(algo=smeshBuilder.MG_Tetra, geom=group_3d)
    algo_3d.Parameters().SetGradation(1.2)

mesh_tet.Compute()

# Build mesh groups on the reference mesh
# ---------------------------------------
mesh_inlet   = mesh_tet.Group(inlet)
mesh_outlet  = mesh_tet.Group(outlet)
mesh_up      = mesh_tet.Group(up)
mesh_down    = mesh_tet.Group(down)
mesh_left    = mesh_tet.Group(left)
mesh_right   = mesh_tet.Group(right)
mesh_cylinders2d = mesh_tet.Group(cylinders2d)

# Generate the coarse mesh
# ------------------------
coarse_grid = mesher.Mesh(gros_geo)
mesh_1d = coarse_grid.Segment()
mesh_1d.NumberOfSegments(1)
algo_2d = coarse_grid.Triangle(algo=smeshBuilder.MEFISTO)
algo_3d = coarse_grid.Tetrahedron(algo=smeshBuilder.MG_Tetra)
coarse_grid.Compute()

# Build mesh groups on the coarse mesh
# ------------------------------------
coarse_inlet   = coarse_grid.Group(gros_inlet)
coarse_outlet  = coarse_grid.Group(gros_outlet)
coarse_up      = coarse_grid.Group(gros_up)
coarse_down    = coarse_grid.Group(gros_down)
coarse_right   = coarse_grid.Group(gros_right)
coarse_left    = coarse_grid.Group(gros_left)

# Export the coarse mesh into a MED file
med_3d_coarse = os.path.join(med_files, "coarse_grid.med")
coarse_grid.ExportMED( med_3d_coarse )

# Build 2D mesh groups
# --------------------
groups_2d_mesh = {}
for face in groups_2d:
    mesh_2d = mesh_tet.Group(face)
    copy_2d = mesher.CopyMesh(mesh_2d, face.GetName())
    face_id = face.GetSubShapeIndices()[0]
    groups_2d_mesh[face_id] = [copy_2d.GetMesh(), mesh_2d, 0]
    faces = copy_2d.GetElementsByType(smeshBuilder.FACE)
    name=face.GetName()
    copy_2d.MakeGroupByIds(name,smeshBuilder.FACE, faces)

# Export of each fine mesh into a MED file
# ----------------------------------------
split_all = []
for tetra in groups_3d:
    copies = []
    tetra_mesh = mesh_tet.Group(tetra)
    copy_3d = mesher.CopyMesh(tetra_mesh, tetra.GetName())
    copies.append(copy_3d.GetMesh())
```

```
    for face in geompy.SubShapeAll(tetra, geompy.ShapeType["FACE"]):
        face_id = geompy.GetSubShapeID(ref_geo, face)
        cp, ori, count = groups_2d_mesh[face_id]
        copies.append(cp)
        groups_2d_mesh[face_id] = [cp, ori, count+1]
    name = tetra.GetName()
    tetra_conc = mesher.Concatenate(copies, True, True, 1e-05, False, name)
    med_file = os.path.join(med_files, name+".med")
    split_name = name
    split_med  = med_file
    split_groups = []
    groups = tetra_conc.GetGroups(smeshBuilder.SMESH.FACE)
    for group in groups:
        split_groups.append(group)
    tetra_conc.ExportMED( med_file )
    split_all.append([split_med, split_name, split_groups])

# Remove internal triangular faces on the reference mesh
# -----------------------------------------------------
for key in groups_2d_mesh.keys():
    cp, ori, count = groups_2d_mesh[key]
    if count == 2:
        list_ids = ori.GetIDs()
        mesh_tet.RemoveElements(list_ids)
    mesh_tet.RemoveGroup(ori)

# export the reference mesh
# -------------------------
med_3d_ref = os.path.join(med_files, "reference.med")
mesh_tet.ExportMED(med_3d_ref)
```

A.2 An example of data file of *TrioCFD*

We present an example of data files executed by *TrioCFD* for solving local problems. As mentioned in Chapter 5, these data files are generated automatically by executing a Python program. Data files in the format .data are text files in which we specify important simulation parameters. To declare an object, we write the command in the form: "object_type name". As indicated in the second line of the example below, we declare "pb" as an object of type "Pb_Hydraulique". To solve a problem or to read a file, the command is in the form: "action parameter1 ... parameterN". For example, in order to read the "tetra_10.med" file, we use the command "Lire_med family_names_from_group_names 10 tetra_10 tetra_10.med" where "Lire_med" is the action followed by several parameters. In data files, we use { } to add nested parameters and use the keyword "fin" or "end" to indicate the end of the file. Texts between two # are comments and not executed by *TrioCFD*.

The data file presented below is based on the original data file of *TrioCFD* with several modifications. The commands "dim 0" and "nom_bord Bord_30" indicate the multiscale basis functions to be computed. The value of "dim" equals to $0, \cdots, s-1$ with s the number of weighting functions defined by (3.7) and (3.203). As described in local problems (3.39), multiscale basis functions are constructed associated to faces of the coarse mesh. Thus "nom_bord" indicates the face associated to which the multiscale basis functions is computed. In this example, the face is "Bord_30". Cl_cellule is the new boundary condition defined in this thesis to stand for the integral type boundary condition in local problems.

```
dimension 2
Pb_Hydraulique pb
Domaine 10
# read the .med file #
```

```
Lire_med family_names_from_group_names 10 tetra_10 tetra_10.med

VEFPreP1B dis
# discretization of pressure P0 or P1 #
Read dis { P0 }

Scheme_euler_implicit sch
Read sch
{
   tinit 0.
   tmax 1e-1
   # dt_min 0.00001 #
   dt_max 10.
   dt_impr 0
   periode_sauvegarde_securite_en_heures 23
   # dt_sauv 100. #
   seuil_statio 1.e-6
   solveur implicite
   {
      seuil_convergence_implicite 1e-5
      # solver in the prediction step #
      solveur gmres { diag seuil 1e-30 nb_it_max 5 }
   }
   facsec 1
   facsec_max 20
}

Fluide_Incompressible fluide
Read fluide
{
   mu       Champ_Uniforme 1 1.
   rho      Champ_Uniforme 1 1.
}

Associate pb 10
Associate pb sch
Associate pb fluide
Discretize pb dis

Read pb
{
  Navier_Stokes_standard
  {
    solveur_pression petsc Cholesky { }
    # Stokes problem when convection is neglected #
    convection { negligeable }
    diffusion { }

    # indicate the basis function to solve #
    dim 0
    nom_bord Bord_30

    initial_conditions
    {
      vitesse Champ_Uniforme 2 0.0 0.0
    }

    boundary_conditions
    {
      Bord_30 Cl_cellule
      Bord_31 Cl_cellule
      Bord_32 Cl_cellule
      Circle  paroi_fixe
```

```
    }
  }

  # set parameters for the post-processing #
  Post_processing
  {
    format LATA
      fields dt_post 1e+10
    {
      vitesse faces
      pression elem
    }
  }

  # export solutions in .sauv format #
    Sauvegarde_simple binaire test.sauv
}

Solve pb
End
```

A.3 A Python script for visualization in *VisIt*

As presented in section 5.4, *VisIt* is used at the end of the multiscale simulation chain to visualize fine-scale solutions. We give an example of the Python script used to launch *VisIt* via the command line interface (CLI).

VisIt can manipulate easily LATA files produced by *TrioCFD*. Since the reconstruction of fine-scale solutions were carried out locally on each coarse element, we obtain as many LATA files as the number of coarse elements. In the loop over LATA files, we first define and open the "database" to read one LATA file. Then we specify the physical field to be visualized in the function "AddPlot()". For example, "VITESSE_FACES_dom_dual_magnitude" and "PRESSURE_ELEM_dom" stands respectively for the velocity computed in face and the pressure computed in elements. Then the physical field is visualized using "DrawPlots()".

Using "View2DAttributes()" and "PseudocolorAttributes()", we specify common parameters for the visualization of the physical field, i.e. the velocity in this example. We visualize velocity in the entire domain Ω^{ε} or only in the regions of interest and we specify this domain in "windowCoords". It is necessary to fix the "min", "max" values and the "colorTableName" for the visualization of velocity in each LATA file. Some annotations, such as "legendFlag", "userInfoFlag" and so on are not necessary for the visualization.

At the end, we export the visualization into a file of format PNG or JPEG. It is important to delete the visualization and close the "database". Then we read the next LATA file and export the visualization of velocity into another PNG file.

```python
import sys
number_of_lATA  = 512
for i in xrange(0, number_of_lATA):
  database = "localhost:../../bases_locales/tetra_"+str(i)+"_post.lata"
  OpenDatabase(database, 0)
  AddPlot("Pseudocolor", "VITESSE_FACES_dom_dual_magnitude", 1, 1)
  DrawPlots()

  # Begin spontaneous state
  View2DAtts = View2DAttributes()
  View2DAtts.windowCoords = (-0.5, 1.5, 0., 1.) #(x1, x2, y1, y2)
  View2DAtts.viewportCoords = (0.2, 0.95, 0.15, 0.95)
  View2DAtts.fullFrameActivationMode = View2DAtts.Auto  # On, Off, Auto
  View2DAtts.fullFrameAutoThreshold = 100
```

```
View2DAtts.xScale = View2DAtts.LINEAR  # LINEAR, LOG
View2DAtts.yScale = View2DAtts.LINEAR  # LINEAR, LOG
View2DAtts.windowValid = 1
SetView2D(View2DAtts)
# End spontaneous state

#set attributs for Pseudocolors
PseudocolorAtts = PseudocolorAttributes()
PseudocolorAtts.legendFlag = 0
PseudocolorAtts.minFlag = 1
PseudocolorAtts.min = 0
PseudocolorAtts.maxFlag = 1
PseudocolorAtts.max = 0.3106
PseudocolorAtts.centering = PseudocolorAtts.Natural  # Natural, Nodal,
  Zonal
PseudocolorAtts.colorTableName = "hot_desaturated"
SetPlotOptions(PseudocolorAtts)

# set annotations
AnnotationAtts = AnnotationAttributes()
AnnotationAtts.userInfoFlag = 0
AnnotationAtts.databaseInfoFlag = 0
SetAnnotationAttributes(AnnotationAtts)

# set png
sw = SaveWindowAttributes()
sw.outputToCurrentDirectory = 1
sw.outputDirectory = "."
sw.fileName = "visit"
sw.family = 1
sw.format = sw.PNG  # JPEG,PNG,POSTSCRIPT, ...
sw.width = 1024
sw.height = 1024
sw.screenCapture = 0
sw.saveTiled = 0
sw.quality = 80
sw.progressive = 0
sw.binary = 0
sw.stereo = 0
sw.compression = sw.PackBits  # None, PackBits, Jpeg, Deflate
sw.forceMerge = 0
sw.resConstraint = sw.ScreenProportions  # NoConstraint, EqualWidthHeight,
  ScreenProportions
sw.advancedMultiWindowSave = 0
SetSaveWindowAttributes(sw)
SaveWindow()

# close the data
DeleteAllPlots()
CloseDatabase(database)
```

Appendix B

Solution of local problems in high-order multiscale methods

In Chapter 4, we presented the discretization of local problems in high-order Crouzeix-Raviart multiscale methods and obtained system (4.57). We expect to solve this system using the $\mathbb{P}_2/\mathbb{P}_1$ finite element. Since this finite element is not available in *TrioCFD*, we solved system (4.57) using the $\mathbb{P}_2/\mathbb{P}_1$ finite element in *Freefem++*. In this appendix, we first prove the existence and uniqueness of A solution to system (4.57). Then we present some multiscale basis functions which are solutions of local problems (3.191)–(3.192).

B.1 The well-posedness of discrete local problems

Now let us prove the existence and uniqueness of a solution to system (4.57). Assuming that $(\mathbf{U}_h, \mathbf{P}_h, \mathbf{\Lambda}_h, \boldsymbol{\eta}_h, \mathbf{K}_h)$ satisfies that

$$\begin{aligned} \mathbf{A}_h\mathbf{U}_h + \mathbf{B}_h^t\mathbf{P}_h + \mathbf{D}_h^t\mathbf{\Lambda}_h + \mathbf{S}_h^t\boldsymbol{\eta}_h &= \mathbf{0} && \text{(B.1)}\\ \mathbf{B}_h\mathbf{U}_h + \mathbf{Q}_h^t\mathbf{K}_h &= \mathbf{0} && \text{(B.2)}\\ \mathbf{D}_h\mathbf{U}_h &= \mathbf{0} && \text{(B.3)}\\ \mathbf{Q}_h\mathbf{P}_h &= \mathbf{0} && \text{(B.4)}\\ \mathbf{S}_h\mathbf{U}_h &= \mathbf{0} && \text{(B.5)} \end{aligned}$$

We want to prove that $\mathbf{U}_h = \mathbf{0},\ \mathbf{P}_h = \mathbf{0},\ \mathbf{\Lambda}_h = \mathbf{0}$ and $\boldsymbol{\eta}_h = \mathbf{0}$. Now let us first prove that $\mathbf{U}_h = \mathbf{0}$. Multiplying (B.1) by $\mathbf{U}_h$ yields

$$(\mathbf{A}_h\mathbf{U}_h, \mathbf{U}_h) + (\mathbf{P}_h, \mathbf{B}_h\mathbf{U}_h) + (\mathbf{\Lambda}_h, \mathbf{D}_h\mathbf{U}_h) + (\boldsymbol{\eta}_h, \mathbf{S}_h\mathbf{U}_h) = 0 \tag{B.6}$$

Making use of (B.2), it is easy to see that

$$(\mathbf{P}_h, \mathbf{B}_h\mathbf{U}_h) = -\left(\mathbf{P}_h, \mathbf{Q}_h^T\mathbf{K}_h\right) = -\left(\mathbf{Q}_h\mathbf{P}_h, \mathbf{K}_h\right)$$

then (B.4) implies that

$$(\mathbf{Q}_h\mathbf{P}_h, \mathbf{K}_h) = 0$$

Equation (B.3) implies that

$$(\mathbf{\Lambda}_h, \mathbf{D}_h\mathbf{U}_h) = 0$$

Equation (B.5) implies that

$$(\boldsymbol{\eta}_h, \mathbf{S}_h\mathbf{U}_h) = 0$$

Finally (B.6) becomes

$$(\mathbf{A}_h \mathbf{U}_h, \mathbf{U}_h) = 0$$

This implies that $\nabla \mathbf{U}_h = 0$ and thus $\mathbf{U}_h$ is constant. Taking advantage of equation (B.5), we conclude that $\mathbf{U}_h = \mathbf{0}$.

In this case, (B.1)–(B.5) reduces to

$$\mathbf{B}_h^t \mathbf{P}_h + \mathbf{D}_h^t \boldsymbol{\Lambda}_h + \mathbf{S}_h^t \boldsymbol{\eta}_h = \mathbf{0} \tag{B.7}$$

$$\mathbf{Q}_h^t \mathbf{K}_h = \mathbf{0} \tag{B.8}$$

$$\mathbf{Q}_h \mathbf{P}_h = \mathbf{0} \tag{B.9}$$

It is straightforward to deduce from (B.8) that $\mathbf{K}_h = \mathbf{0}$. It remains to prove that $\mathbf{P}_h = \mathbf{0}$, $\boldsymbol{\Lambda}_h$ and $\boldsymbol{\eta}_h = \mathbf{0}$. For a velocity node $\boldsymbol{x}_f$ located in the interior of the domain, (B.7) reduces to

$$\mathbf{B}_h^t \mathbf{P}_h + \mathbf{S}_h^t \boldsymbol{\eta}_h = \mathbf{0}$$

Making use of $\mathbf{S}_h^t$ defined by (4.47), in the control volume of velocity ω_f shown in Figure 4.2 and Figure 4.7,

$$\left(\mathbf{B}_h^t \mathbf{P}_h\right)_{\omega_f} = \operatorname{meas}(\omega_f)\, \boldsymbol{\eta}_h \tag{B.10}$$

where

$$\operatorname{meas}(\omega_f) = \frac{\operatorname{meas}(k_1) + \operatorname{meas}(k_2)}{3} \tag{B.11}$$

With the Crouzeix-Raviart finite element If unknowns in local problems is discretized by the Crouzeix-Raviart finite element shown in Figure 4.2, then the pressure is constant in each element. The discretization of the gradient of pressure is presented in subsection 4.2.2. Using the coefficient of the matrix $\mathbb{C}_h$, the gradient of pressure on the face f shown in Figure 4.2 is computed by

$$\left(\mathbf{B}_h^t \mathbf{P}_h\right)_{\omega_f} = (p_1 - p_2)\, \boldsymbol{S}_{k_1}^f$$

We recall that $\boldsymbol{S}_{k_1}^f + \boldsymbol{S}_{k_2}^f = \mathbf{0}$. In the equality above, we choose $\boldsymbol{S}_{k_1}^f$ as the conventional orientation in k_1 and k_2.

On combining (B.10), we deduce that $\boldsymbol{\eta}_h$ is collinear with $\boldsymbol{S}_{k_1}^f$. Besides, the same result is obtained for the two other faces m and n of the element k_1. This result implies that $\boldsymbol{\eta}_h$ is also collinear with both $\boldsymbol{S}_{k_1}^m$ and $\boldsymbol{S}_{k_1}^n$. In conclusion, $\boldsymbol{\eta}_h = \mathbf{0}$ which implies that $p_1 = p_2$. Using the same arguments for any internal face f of the mesh $\mathcal{T}_h(T)$, it is easy to verify that p has the same value on all elements of $\mathcal{T}_h(T)$. In other words, p is in the space $\mathbf{P}_0(T)$. Moreover, making use of equation (B.9), it is straightforward to deduce that $\mathbf{P}_h = \mathbf{0}$. On combining $\boldsymbol{\Lambda}_h = \mathbf{0}$, (B.7) implies that $\boldsymbol{\Lambda}_h = \mathbf{0}$.

With the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element If velocity and pressure in local problems is discretized by the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ element shown in Figure 4.7, then the pressure is in the $\mathbf{P}_1(k)$ space in each element $k \in \mathcal{T}_h(T)$. As shown in Figure 4.8, for the internal f, we denote by $\boldsymbol{\tau}_{S_1 S_2}$ the vector collinear with $[S_1 S_2]$ and of $\operatorname{meas}(S_1 S_2)$.

We compute $\left(\mathbf{B}_h^t\mathbf{P}_h\right)_{\omega_f}$ by

$$\left(\mathbf{B}_h^t\mathbf{P}_h\right)_{\omega_f} = \frac{\mathrm{meas}\,(k_1)}{3}\left(\nabla\mathbf{P}_h\right)|_{k_1} + \frac{\mathrm{meas}\,(k_2)}{3}\left(\nabla\mathbf{P}_h\right)|_{k_2}$$

We multiply the equality above by the vector $\boldsymbol{\tau}_{S_1S_2}$ and we obtain

$$\left(\mathbf{B}_h^t\mathbf{P}_h\right)_{\omega_f}\cdot\boldsymbol{\tau}_{S_1S_2} = \frac{\mathrm{meas}\,(k_1)}{3}\left(\nabla\mathbf{P}_h\right)|_{k_1}\cdot\boldsymbol{\tau}_{S_1S_2} + \frac{\mathrm{meas}\,(k_2)}{3}\left(\nabla\mathbf{P}_h\right)|_{k_2}\cdot\boldsymbol{\tau}_{S_1S_2} \tag{B.12}$$

Besides, the definition of the gradient of pressure implies that

$$\begin{aligned}\left(\nabla\mathbf{P}_h\right)|_{k_1}\cdot\boldsymbol{\tau}_{S_1S_2} &= P_{S_2} - P_{S_1}\\ \left(\nabla\mathbf{P}_h\right)|_{k_2}\cdot\boldsymbol{\tau}_{S_1S_2} &= P_{S_2} - P_{S_1}\end{aligned}$$

Using this equality, (B.12) can be written as

$$\left(\mathbf{B}_h^t\mathbf{P}_h\right)_{\omega_f}\cdot\boldsymbol{\tau}_{S_1S_2} = \left(\frac{\mathrm{meas}\,(k_1)}{3} + \frac{\mathrm{meas}\,(k_2)}{3}\right)\left(\nabla\mathbf{P}_h\right)|_{k_1} \tag{B.13}$$

Meanwhile, multiplying (B.10) by $\boldsymbol{\tau}_{S_1S_2}$, we obtain

$$\left(\mathbf{B}_h^t\mathbf{P}_h\right)_{\omega_f}\cdot\boldsymbol{\tau}_{S_1S_2} = \mathrm{meas}\,(\omega_f)\,\boldsymbol{\eta}_h\cdot\boldsymbol{\tau}_{S_1S_2} \tag{B.14}$$

Combining (B.13) and (B.14), we conclude that

$$\left(\frac{\mathrm{meas}\,(k_1)}{3} + \frac{\mathrm{meas}\,(k_2)}{3}\right)\left(\nabla\mathbf{P}_h\right)|_{k_1} = \mathrm{meas}\,(\omega_f)\,\boldsymbol{\eta}_h\cdot\boldsymbol{\tau}_{S_1S_2}$$

Making use of (B.11), the equality above reduces to

$$\left(\nabla\mathbf{P}_h\right)|_{k_1}\cdot\boldsymbol{\tau}_{S_1S_2} = \boldsymbol{\eta}_h\cdot\boldsymbol{\tau}_{S_1S_2}$$

Using the same arguments, it is straightforward to verify that this equality holds true for $\boldsymbol{\tau}_{S_1S_{op1}}$ and $\boldsymbol{\tau}_{S_2S_{op1}}$ in the element k_1. In this case, it is easy to deduce that

$$\left(\nabla\mathbf{P}_h\right)|_{k_1} = \boldsymbol{\eta}_h \tag{B.15}$$

Proceeding in this method, for any internal face f in the mesh $\mathcal{T}_h(T)$, we conclude that (B.15) is true for any element of the mesh $\mathcal{T}_h(T)$ and thus $\mathbf{P}_h$ is in the $\mathbb{P}_1(T)$ space. Combining (B.9), we conclude that $\mathbf{P}_h = \mathbf{0}$. Consequently, (B.10) implies that $\boldsymbol{\eta}_h = \mathbf{0}$ and then (B.7) reveals that $\boldsymbol{\Lambda}_h = \mathbf{0}$.

On the mesh $\mathcal{T}$, there are some elements which contain two faces located on the boundary. When the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ element is used to discretize local problems, it is recommended to use the function *VeryfierCoin* in *TrioCFD* to split this kind of elements into three smaller elements. This operation is illustrated in Figure 4.10. After this operation, elements located on the boundary of $\mathcal{T}$ contains only one face on the boundary but two internal faces. For these two internal faces, we deduce (B.15) proceeding in the same way as above.

Consequently, we have proved the existence and the uniqueness of a solution to system (B.1)–(B.5) for both the Crouzeix-Raviart finite element and the $\mathbb{P}_1$-nonconforming/$\mathbb{P}_1$ finite element.

B.2 Multiscale basis functions

In two dimensions, we solve system (3.191)–(3.192) on the fine mesh shown in Figure B.1 using the $\mathbb{P}_2/\mathbb{P}_1$ finite element in *Freefem++*. We choose weighting functions defined by (3.203) and we recall below:

$$\begin{cases} s = 4: \; \boldsymbol{\omega}_{E,1} = \boldsymbol{e}_1, \; \boldsymbol{\omega}_{E,2} = \boldsymbol{e}_2, \; \boldsymbol{\omega}_{E,3} = \boldsymbol{n}_E \psi_E, \; \boldsymbol{\omega}_{E,4} = \boldsymbol{\tau}_E \phi_E. \\ t = 3: \; \varpi_{T,1} = 1, \; \varpi_{T,2} = x, \; \varpi_{T,3} = y. \\ r = 2: \; \boldsymbol{\varphi}_{T,1} = \boldsymbol{e}_1, \; \boldsymbol{\varphi}_{T,2} = \boldsymbol{e}_2. \end{cases} \tag{B.16}$$

where $\{\boldsymbol{e}_1, \; \boldsymbol{e}_2\}$ is the canonical basis of $\mathbb{R}^2$ and $\boldsymbol{n}_E$ and $\boldsymbol{\tau}_E$ are respectively unit vectors normal and tangent to the face E. Besides, $\psi_E \in \mathbb{P}_1(E)$ and $\phi_E \in \mathbb{P}_1(E)$ which verify that $\int_{E\cap\Omega^\varepsilon} \psi_E = 0$ and $\int_{E\cap\Omega^\varepsilon} \phi_E = 0$.

On the coarse element T, we obtain 14 multiscale basis functions including $(\boldsymbol{\Psi}_{T,1}, \pi_{T,1})$, $(\boldsymbol{\Psi}_{T,2}, \pi_{T,2})$ and $(\boldsymbol{\Phi}_{F,1}, \pi_{F,1}), \cdots, (\boldsymbol{\Phi}_{F,4}, \pi_{F,4})$ for any $F \in \mathcal{E}(T)$. We recall that the dimension of $\mathcal{E}(T)$ is 3 for a triangular coarse element in two dimensions.

As shown in Figure B.2, the accompanying pressure $\pi_{T,1}$ and $\pi_{T,2}$ are both zero in the entire domain. Figure B.3 reveals the multiscale basis functions $\boldsymbol{\Psi}_{T,1}$ and $\boldsymbol{\Psi}_{T,2}$. On combining the pressure contour, it is easy to deduce that $\boldsymbol{\Psi}_{T,1}$ and $\boldsymbol{\Psi}_{T,2}$ are both quadratic polynomials on T.

Figures B.4 to B.5 present the multiscale basis functions $\boldsymbol{\Phi}_{F,1}, \cdots, \boldsymbol{\Phi}_{F,4}$ associated to the boundary F. As shown in Figures B.6 to B.7, the accompanying pressure $\pi_{E,1}$ and $\pi_{E,2}$ are zero in the entire domain while $\pi_{E,3}$ and $\pi_{E,4}$ are not zero at all. On combining the pressure contour, we deduce that $\boldsymbol{\Phi}_{F,1}$ and $\boldsymbol{\Phi}_{F,2}$ are both quadratic polynomials whereas $\boldsymbol{\Phi}_{F,3}$ and $\boldsymbol{\Phi}_{F,4}$ are not.

In conclusion, among the 14 multiscale basis functions computed on T, $\boldsymbol{\Psi}_{T,1}$ and $\boldsymbol{\Psi}_{T,2}$ are both quadratic polynomials. For any boundary $F \in \mathcal{E}(T)$, $\boldsymbol{\Phi}_{F,1}$ and $\boldsymbol{\Phi}_{F,2}$ are quadratic polynomials while $\boldsymbol{\Phi}_{F,3}$ and $\boldsymbol{\Phi}_{F,4}$ are not.

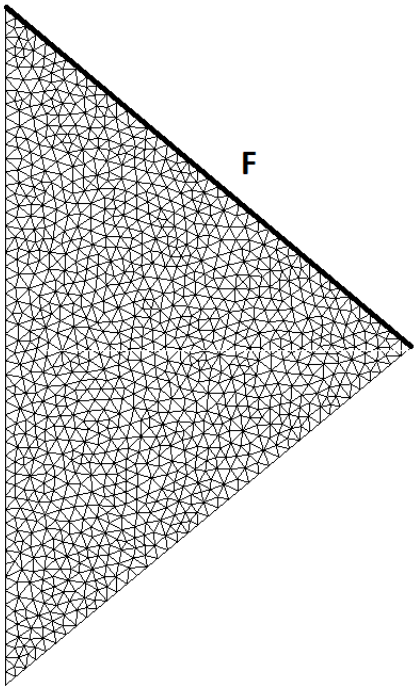
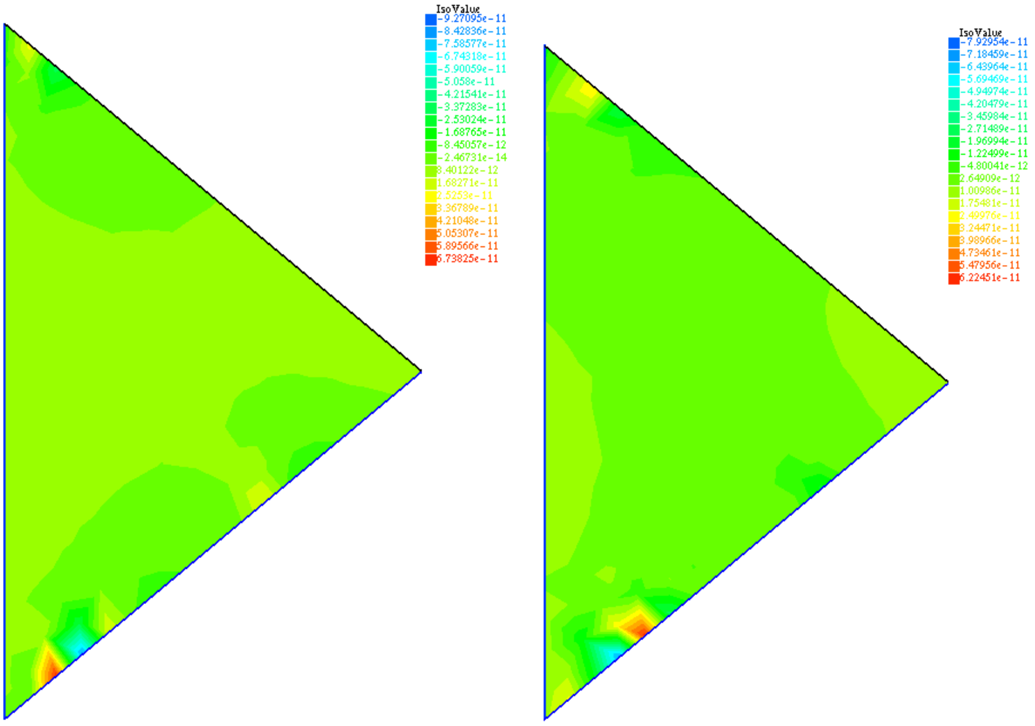
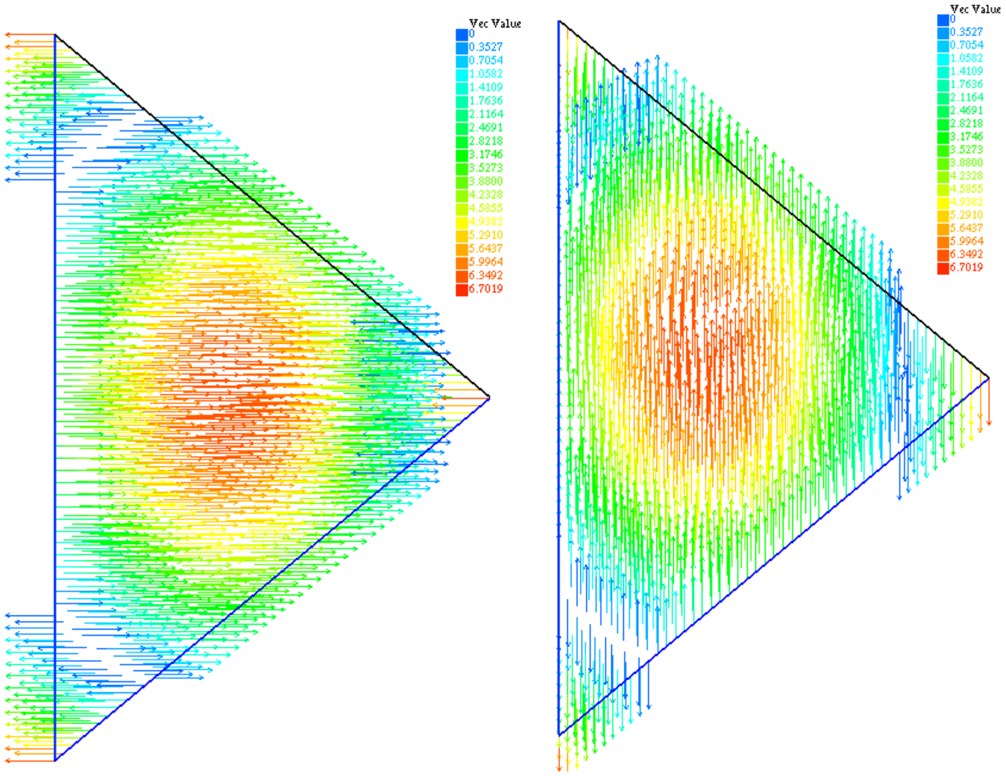


FIGURE B.1: Mesh on the coarse element T



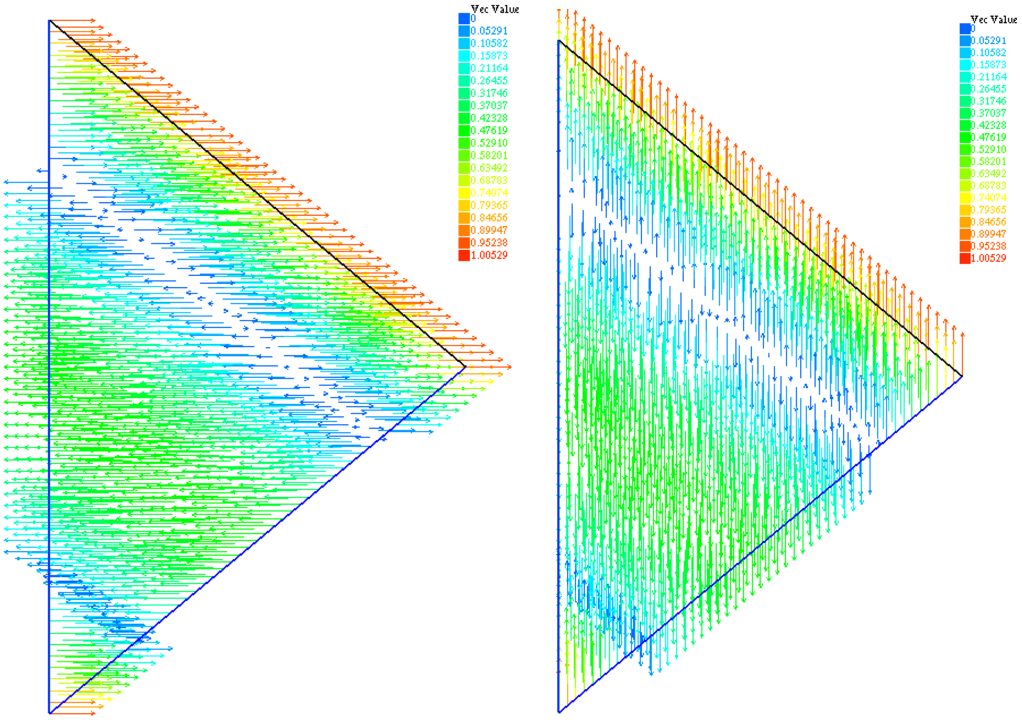
(A) Contour $\pi_{T,1}$ (B) Contour $\pi_{T,2}$

FIGURE B.2: Pressure contour $\pi_{T,1}$ and $\pi_{T,2}$



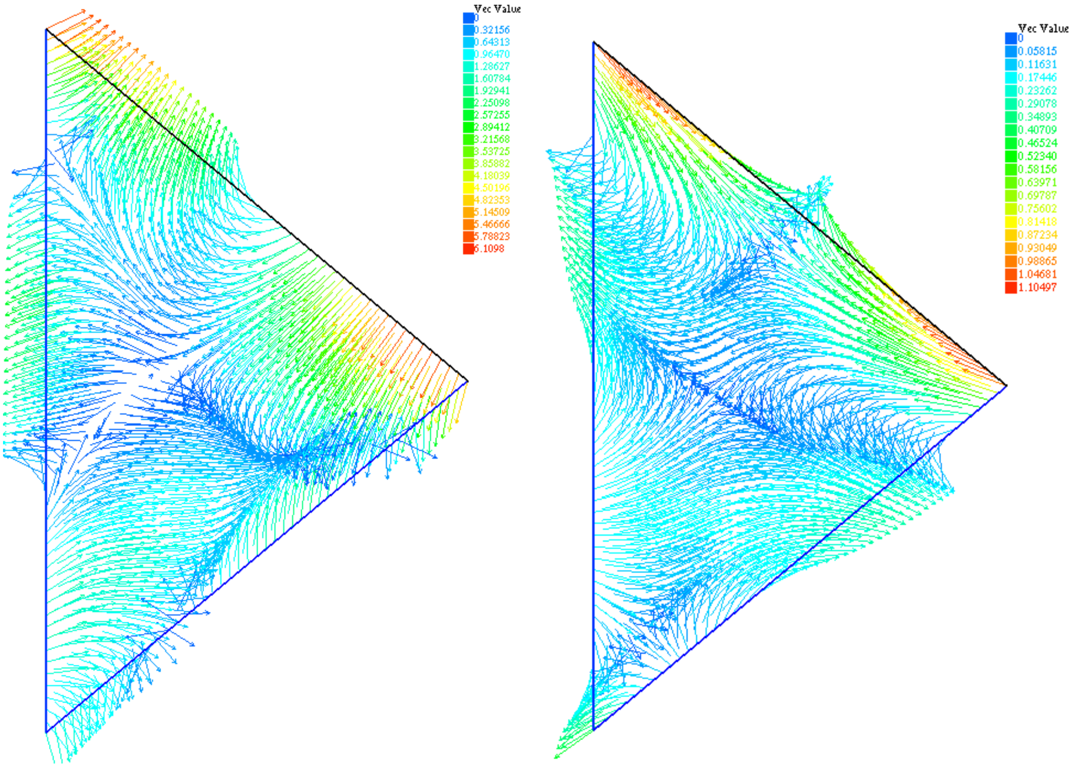
(A) Vector $\boldsymbol{\Psi}_{T,1}$ (B) Vector $\boldsymbol{\Psi}_{T,2}$

FIGURE B.3: Multiscale basis functions $\boldsymbol{\Psi}_{T,1}$ and $\boldsymbol{\Psi}_{T,2}$



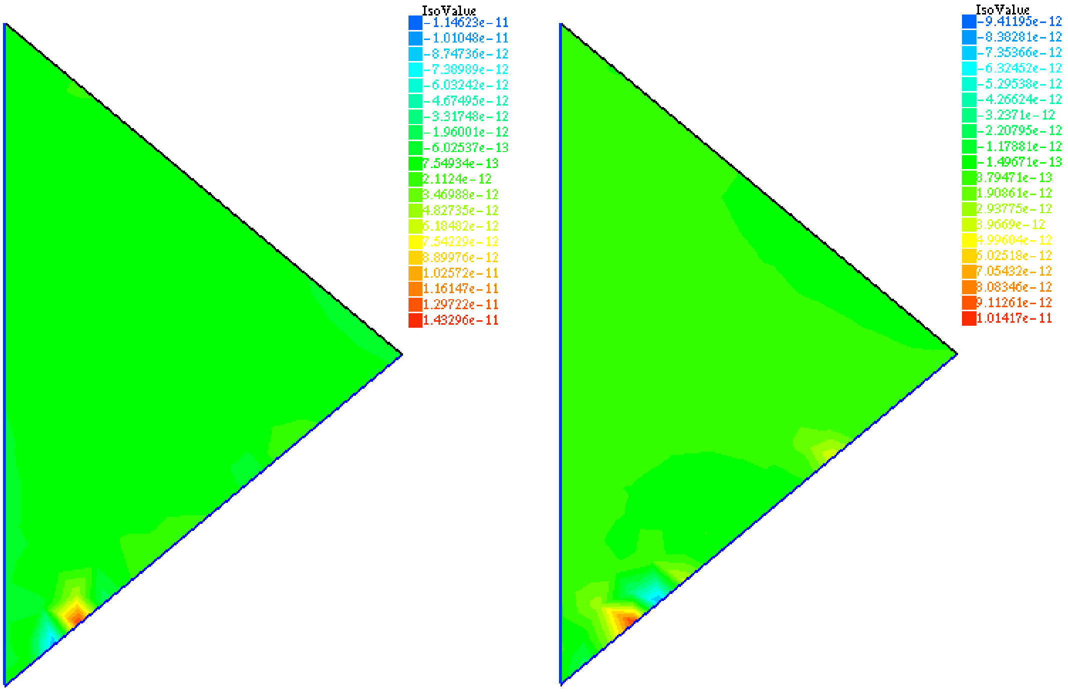
(A) Vector $\mathbf{\Phi}_{F,1}$ (B) Vector $\mathbf{\Phi}_{F,2}$

FIGURE B.4: Multiscale basis functions $\mathbf{\Phi}_{F,1}$ and $\mathbf{\Phi}_{F,2}$



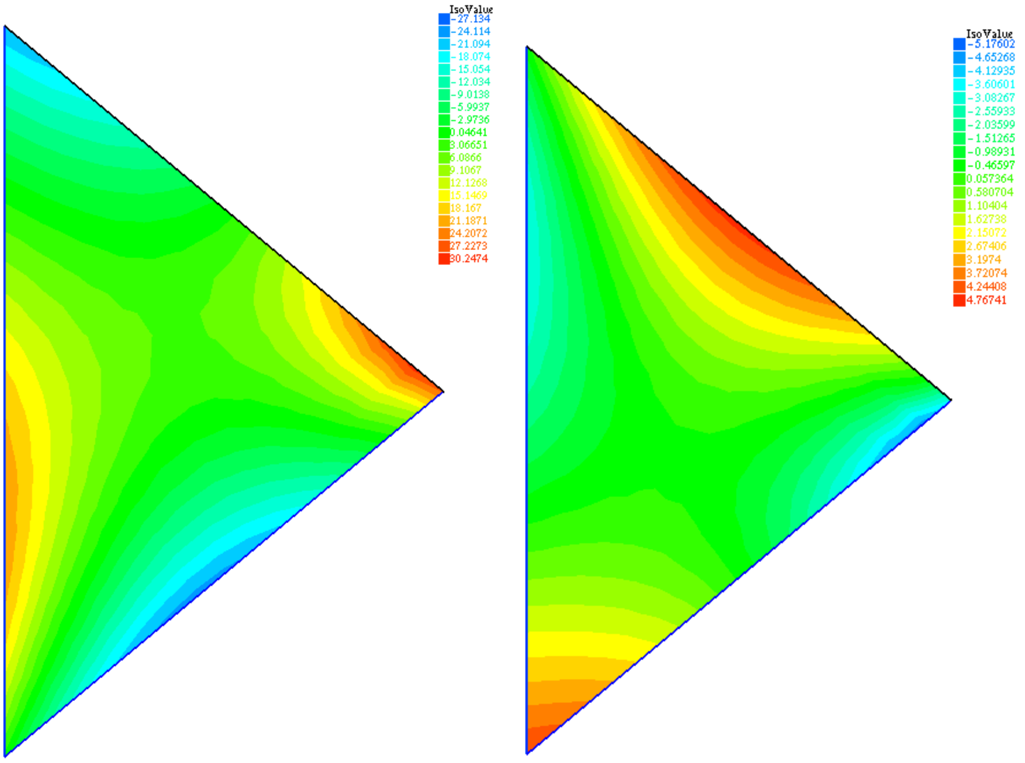
(A) Vector $\mathbf{\Phi}_{F,3}$ (B) Vector $\mathbf{\Phi}_{F,4}$

FIGURE B.5: Multiscale basis functions $\mathbf{\Phi}_{F,3}$ and $\mathbf{\Phi}_{F,4}$



(A) Contour $\pi_{F,1}$ (B) Contour $\pi_{F,2}$

FIGURE B.6: Pressure contour $\pi_{F,1}$ and $\pi_{F,2}$



(A) Contour $\pi_{F,3}$ (B) Contour $\pi_{F,4}$

FIGURE B.7: Pressure contour $\pi_{F,3}$ and $\pi_{F,4}$

Appendix C

Résumé en français

Le cœur d'un réacteur nucléaire est un milieu très hétérogène encombré de nombreux obstacles solides (crayons combustibles, grilles de mélange etc.) aux différentes échelles. Pour modéliser les écoulements dans le cœur des réacteurs nucléaires, les codes les plus utilisés sont les codes thermohydrauliques à l'échelle système, tel que CATHARE, RELAP5, ATHLET etc. Ces codes sont basés sur des méthodes moyennées et les équations unidimensionnelles de masse, de quantité de mouvement, d'énergie etc. Mais ces codes ne peuvent pas bien représenter les caractéristiques tridimensionnelles dans le cœur des réacteurs nucléaires.

La CFD (Computational Fluid Dynamics en anglais) est un moyen approprié qui peut bien modéliser les phénomènes tridimensionnelles très complexes. Les études CFD des écoulements dans un réacteur à l'échelle réduite montrent que les phénomènes thermohydrauliques à l'échelle macroscopique sont directement impactés par les phénomènes à l'échelle microscopique. Les phénomènes dans le cœur des réacteurs nucléaires sont donc des problèmes multi-échelles. Toutefois la simulation CFD de l'écoulement dans le cœur complet exige un maillage extrêmement fin et nécessite une quantité de ressources de calculs plus grande que les ressources informatiques actuelles.

Dans la littérature, il y a un grand nombre de techniques (model reduction method en anglais) qui ont été développées pour résoudre les problèmes multi-échelles sur un maillage grossier, en capturant les caractéristiques microscopiques les plus importantes. Ces méthodes ont toutes emprunté l'idée de la théorie de l'homogénéisation [16, 29, 86, 98]. Parmi ces méthodes, on trouve les méthodes d'éléments finis multi-échelles [61, 87, 88], les méthodes de volumes finis multi-échelles [72, 96, 97], les méthodes multi-échelles variationnelles [91, 92] ou les méthodes multi-échelles hétérogènes [7, 57, 58, 83] etc. Cette thèse est consacrée au développement de méthodes d'éléments finis multi-échelles pour simuler les écoulements incompressibles dans un milieu hétérogène.

La méthode d'éléments finis multi-échelles s'effectue en quatre étapes. La première étape est de générer un maillage grossier et des maillages fins. Le maillage grossier ne contient pas d'obstacles et les maillages fins sont suffisamment fins pour bien modéliser la frontière des obstacles dans le milieu. La deuxième étape est de construire les fonctions de base multi-échelles par résoudre les problèmes locaux sur les maillages fins. Dans la troisième étape, sur le maillage grossier, on résout un problème grossier qui est discrétisé par une méthode de Galerkin où les fonctions de bases sont les solutions des problèmes locaux. La quatrième étape est de reconstruire les caractéristiques microscopiques des phénomènes physiques sur les maillages fins en utilisant les fonctions de base multi-échelles et les solutions des problèmes grossiers.

Dans cette thèse, dans un premier temps, on implémente la méthode d'éléments finis multi-échelles originellement proposée par [95, 118], où les problèmes locaux sont définis par les équations de Stokes avec des conditions aux limites appropriées. Dans le travail originel, les auteurs ont résolu seulement le problème de Stokes sur le maillage grossier. Dans cette thèse, on propose de résoudre aussi les problèmes d'Oseen et de Navier-Stokes sur le maillage grossier. La méthode de Streamline-Upwind/Petrov-Galerkin (SUPG) [33] est développée pour stabiliser les solutions sur le maillage grossier.

Pour prendre en compte le terme de convection dans la définition des fonctions de base multi-échelles, on peut construire une méthode d'éléments finis multi-échelles où les problèmes locaux sont définis par les équations d'Oseen avec des conditions aux limites appropriées. Cette méthode a été d'abord proposée dans [117] et on a d'abord implémenté les problèmes locaux proposés dans la méthode. Néanmoins, nos simulations numériques montrent qu'il existe des oscillations dans les solutions des problèmes locaux quand la vitesse d'Oseen est relativement grande. Par conséquent, on définit des problèmes locaux différemment et démontre qu'ils sont bien-posés. En utilisant la nouvelle définition, il n'existe plus d'oscillations même aux grandes vitesses d'Oseen. Dans [117], l'auteur a résolu seulement les problèmes d'Oseen sur le maillage grossier. Dans cette thèse, on propose de résoudre non seulement les problèmes d'Oseen mais aussi les problèmes de Navier-Stokes sur le maillage grossier. La méthode SUPG [33] est développée pour stabiliser les solutions sur le maillage grossier.

Dans un deuxième temps, on propose deux idées pour enrichir les deux méthodes multi-échelles presentées précédemment afin d'améliorer la précision des solutions numériques. La première idée est d'enrichir seulement l'espace d'approximation de la vitesse. Pour cet objectif, on propose deux méthodes d'enrichissement : (1) Prendre l'ensemble des fonctions de base définies respectivement par les équations de Stokes et les équations d'Oseen. On espère obtenir une méthode plus précise que la méthode définie par les équations de Stokes ou d'Oseen seules. Pour résoudre le problème d'Oseen sur le maillage grossier, les résultats numériques montrent que la méthode enrichie est plus précise que la méthode définie par les fonctinos de base de Stokes seules mais moins précise que celle définie par les fonctions de base d'Oseen seules. (2) Enrichir l'espace d'approximation de la vitesse en ajoutant des fonctions bulles. On a étendu les fonctions bulles proposées pour les équations de diffusion ou diffusion-advection [54, 102, 113] aux équations de Stokes. Mais nos expériences numériques montrent que l'addition des fonctions bulles n'améliore pas la précision des résultats numériques. Puis nos analyses théoriques montrent que l'espace d'approximation de la pression doit aussi être enrichi, menant à la deuxième idée d'enrichissement.

La deuxième idée est d'enrichir les espaces d'approximation de la vitesse et de la pression à l'aide des fonctions de poids, qui sont définies par les polynômes de degré plus élevé que précedemment. On indique comment choisir les fonctions de poids et les degrés des polynômes dans les espaces d'approximation de la pression et de la vitesse. On définit les nouveaux problèmes locaux par les équations de Stokes ou Oseen. Par conséquent, on obtient une méthode innovante d'éléments finis multi-échelles qui est plus générale que toutes les méthodes présentées précédemment. En faisant varier l'ordre des polynômes dans la définition des fonctions de poids, on peut trouver un bon compromis entre la précision de la méthode et le coût des calculs. Nos expériences numériques montrent que cette méthode multi-échelle améliore significativement la précision de la vitesse et de la pression.

Une chaîne de simulation multi-échelles *SALOME-TrioCFD-VisIt* est construite pour simuler des écoulements dans des milieux hétérogènes de dimension deux et trois. Les maillages nécessaires sont générés par un code industriel *SALOME* [129], en utilisant les algorithmes spécifiques développés dans cette thèse. Les méthodes d'éléments finis multi-échelles sont implémentées et validées dans un code industriel du CEA, nommé *TrioCFD* [139]. La visualisation des solutions fines est réalisée à l'aide du logiciel *VisIt* [142].

Bibliography

[1] J. Aarnes and B. O. Heimsund. “Multiscale discontinuous Galerkin methods for elliptic problems with multiple scales”. In: *Multiscale Methods in Science and Engineering. Lecture Notes in Computational Science and Engineering*. Springer, Berlin, Heidelberg, 2005.

[2] T. Abballe. “Simulation multi-échelle et homogénéisation des matériaux cimentaires”. PhD thesis. École Polytehnique, 2011. URL: https://pastel.archives-ouvertes.fr/pastel-00627899/.

[3] A. Abdulle. “Heterogeneous multiscale method for elliptic problems with multiple scales”. In: *Mathematics of Computation* 81.278 (2012), pp. 687–713.

[4] A. Abdulle and O. Budáč. “A reduced basis finite element heterogeneous multiscale method for Stokes flow in porous media”. In: *Computer Methods in Applied Mechanics and Engineering* 307 (2016), pp. 1–31.

[5] A. Abdulle and O. Budáč. “An adaptive finite element heterogeneous multiscale method for Stokes flow in porous media”. In: *Multiscale Modeling & Simulation* 13.1 (2015), pp. 256–290.

[6] A. Abdulle, O. Budáč, and A. Imboden. “A three-scale offline–online numerical method for fluid flow in porous media”. In: *Journal of Computational Physics* 337 (2017), pp. 175–202.

[7] A. Abdulle et al. “The heterogeneous multiscale method”. In: *Acta Numerica* 21 (2012), pp. 1–87.

[8] F. Abraham, M. Behr, and M. Heinkenschloss. “The effect of stabilization in finite element methods for the optimal boundary control of the Oseen equations”. In: *Finite Elements in Analysis and Design* 41.3 (2004), pp. 229–251.

[9] R. A. Adams and J. Fournier. *Sobolev spaces*. 2nd. Academic Press, 2003.

[10] G. Allaire. “A brief introduction to homogenization and miscellaneous applications”. In: *ESAIM: Proceedings* 37 (2012), pp. 1–49.

[11] G. Allaire. “Homogenization and two-scale convergence”. In: *SIAM Journal on Mathematical Analysis* 23.6 (1992), pp. 1482–1518.

[12] G. Allaire. “Homogenization in porous media, Lecture 2 of CEA-EDF-INRIA school on homogenization”. 2010.

[13] G. Allaire. “Homogenization of the Navier-Stokes equations in open sets perforated with tiny holes II: Non-critical sizes of the holes for a volume distribution and a surface distribution of holes”. In: *Arch. Rational Mech. Anal.* 113.3 (1991), pp. 261–298.

[14] G. Allaire. “Homogenization of the Stokes flow in a connected porous medium”. In: *Asymptotic Analysis* 2.3 (1989), pp. 203–222.

[15] G. Allaire. “Homogenization of the unsteady Stokes equations in porous media”. In: *Progress in partial differential equations: calculus of variations, applications* 267 (1992), pp. 109–123.

[16] G. Allaire. *Shape optimization by the homogenization method.* Springer-Verlag, New York, 2001.

[17] G. Allaire and R. Brizzi. “A multiscale finite element method for numerical homogenization”. In: *Multiscale Modeling and Simulation* 4.3 (2004), pp. 790–812.

[18] G. Allaire and K. E. Ganaoui. “Homogenization of a conductive and radiative heat transfer problem”. In: *Multiscale Modeling & Simulation* 7.3 (2009), pp. 1148–1170.

[19] G. Allaire and Z. Habibi. “Homogenization of a conductive, convective and radiative heat transfer problem in a heterogeneous domain”. In: *SIAM Journal on Mathematical Analysis* 45.3 (2013), pp. 1136–1178.

[20] P.-E. Angeli. “Simulation multi-résolution/multi-échelle de la thermohydraulique des assemblages de réacteur à neutrons rapides”. PhD thesis. École Centrale Paris, 2011. URL: `https://tel.archives-ouvertes.fr/tel-00678241/`.

[21] P. Angot, C. H. Bruneau, and P. Fabrie. “A penalization method to take into account obstacles in incompressible viscous flows”. In: *Numerische Mathematik* 81 (1999), pp. 497–520.

[22] T. Arbogast. “Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems”. In: *SIAM Journal on Numerical Analysis* 42.2 (2004), pp. 576–598.

[23] T. Arbogast. “Numerical subgrid upscaling of two-phase flow in porous media. Lecture Notes in Physics”. In: *Numerical treatment of multiphase flows in porous media* 552 (2000), pp. 35–49.

[24] T. Arbogast and K. Boyd. “Subgrid upscaling and mixed multiscale finite elements”. In: *SIAM Journal on Numerical Analysis* 44.3 (2006), pp. 1150–1171.

[25] T. Arbogast, S. Minkoff, and P. Keenan. “An operator-based approach to upscaling the pressure equation”. In: *Computational Methods in Water Resources XII: Computational Mechanics Publications* (1998), pp. 405–412.

[26] T. Arbogast et al. “A multiscale mortar mixed finite element method”. In: *Multiscale Modeling & Simulation* 6.1 (2007), pp. 319–346.

[27] ATHLET. URL: `https://www.grs.de/en/computer-code-athlet`.

[28] I. Babuška. “The finite element method with Lagrangian multipliers”. In: *Numerische Mathematik* 20.3 (1973), pp. 179–192.

[29] N. Bakhvalov and G. Panasenko. *Homogenization: averaging processes in periodic media. Mathematical problems in the mechanics of composite materials*. Springer Netherlands, 1989.

[30] R. Bavière et al. “A first system/CFD coupled simulation of a complete nuclear reactor transient using CATHARE2 and TRIO-U. Preliminary validation on the Phénix reactor natural circulation test”. In: *Nuclear Engineering and Design* 277 (2014), pp. 124–137.

[31] G. Bonfigli and P. Jenny. “An efficient multi-scale Poisson solver for the incompressible Navier-Stokes equations with immersed boundaries”. In: *Journal of Computational Physics* 228.12 (2009), pp. 4568–4587.

[32] F. Brezzi. “On the existence, uniqueness and approximation of saddle-point problems arising from lagrange multipliers”. In: *ESIAM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* 8.R2 (1972), pp. 129–151.

[33] A. N. Brooks and T. J. R. Hughes. “Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations”. In: *Computer Methods in Applied Mechanics and Engineering* 32.1-3 (1982), pp. 199–259.

[34] Z. Cai. “On the finite volume element method”. In: *Numerische Mathematik* 58 (1991), pp. 713–735.

[35] CATHARE. URL: `https://www.irsn.fr/EN/Research/Scientific-tools/Computer-codes/Pages/T-he-CATHARE2-code-4661.aspx`.

[36] CFX. URL: `https://www.ansys.com/fr-fr/products/fluids/ansys-cfx`.

[37] M. Chandesris. “Modélisation des écoulements turbulents dans les milieux poreux et à l’interface avec un milieu libre”. PhD thesis. Université de Paris VI, 2006. URL: `http://www.theses.fr/2006PA066456`.

[38] R. Chen et al. “Three dimensional thermal hydraulic characteristic analysis of reactor core based on porous media method”. In: *Annuals of Nuclear Energy* 104 (2017), pp. 178 –190.

[39] Y. Chen et al. “A coupled local-global upscaling approach for simulating flow in highly heterogeneous formations”. In: *Advances in Water Resources* 26 (2003), pp. 1041–1060.

[40] Z. Chen and T. Y. Hou. “A mixed multiscale finite element method for elliptic problems with oscillating coefficients”. In: *Mathematics of Computation* 72.242 (2002), pp. 541–576.

[41] Z. Chen et al. “The multiscale finite element method with nonconforming elements for elliptic homogenization problems”. In: *Multiscale Modeling & Simulation* 7.2 (2008), pp. 517–538.

[42] A. J. Chorin. “Numerical solution of the Navier-Stokes equations”. In: *Mathematics of Computation* 22 (1968), pp. 745–762.

[43] J. Chu et al. “Flow based oversampling technique for multiscale finite element methods”. In: *Advances in Water Resources* 31.4 (2008), pp. 599–608.

[44] E. Chung, Y. Efendiev, and C. Lee. “Mixed generalized multiscale finite element methods and applications”. In: *Multiscale Modeling & Simulation* 13.1 (2015), pp. 338–366.

[45] E. Chung et al. “Generalized multiscale finite element methods for problems in perforated heterogeneous domains”. In: *Applicable Analysis* 95.10 (2016), pp. 2254–2279.

[46] E. T. Chung, W. T. Leung, and M. Vasilyeva. “Mixed GMsFEM for second order elliptic problem in perforated domains”. In: *Journal of Computational and Applied Mathematics* 304 (2016), pp. 84–99.

[47] E. T. Chung, M. Vasilyeva, and Y. Wang. “A conservative local multiscale model reduction technique for Stokes flows in heterogeneous perforated domains”. In: *Journal of Computational and Applied Mathematics* 321 (2017), pp. 389–405.

[48] E. T. Chung et al. “Multiscale model reduction for transport and flow problems in perforated domains”. In: *Journal of Computational and Applied Mathematics* 330 (2018), pp. 519–535.

[49] E. T. Chung et al. “Non-local multi-continua upscaling for flows in heterogeneous fractured media”. In: *Journal of Computational Physics* 372 (2018), pp. 22–34.

[50] M. Cicuttin, A. Ern, and S. Lemaire. “A hybrid high-order method for highly oscillatory elliptic problems”. In: *Computational Methods in Applied Mathematics, De Gruyter, In press, 10.1515/cmam-2018-0013* (2018).

[51] D. Cortinovis and P. Jenny. “Iterative Galerkin-enriched multiscale finite-volume method”. In: *Journal of Computational Physics* 277 (2014), pp. 248–267.

[52] M. Crouzeix and P.-A. Raviart. “Conforming and nonconforming finite element methods for solving the stationary Stokes equations I”. In: *Revue française d’automatique, informatique, recherche opérationnell. Mathématique* 7.3 (1973), pp. 33–75.

[53] T. A. Davis. *UMFPACK quick start guide*. 2018.

[54] P. Degond et al. “Crouzeix-Raviart MsFEM with bubble functions for diffusion and advection-diffusion in perforated media”. In: *Communications in Computational Physics* 17.4 (2013), pp. 887–907.

[55] DISTENE. URL: `http://www.distene.fr/`.

[56] V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods: algorithms, theory and parallel implementation*. 2015.

[57] Weinan E. and B. Engquist. “The heterogeneous multiscale methods”. In: *Communications in Mathematical Sciences* 1.1 (2003), pp. 87–132.

[58] Weinan E. et al. “The heterogeneous multiscale method: A review”. In: *Communications in Computational Physics* 2.3 (2007), pp. 367–450.

[59] Y. Efendiev, J. Galvis, and T. Y. Hou. “Generalized Multiscale Finite Element Methods (GMsFEM)”. In: *Journal of Computation Physics* 251 (2013), pp. 116–135.

[60] Y. Efendiev, J. Galvis, and X. H. Wu. “Multiscale finite element methods for high-contrast problems using local spectral basis functions”. In: *Journal of Computational Physics* 230.4 (2011), pp. 937–955.

[61] Y. Efendiev and T. Y. Hou. *Multiscale finite element methods: Theory and applications. Surveys and tutorials in the applied mathematical sciences*. Springer, New York, NY, 2009.

[62] Y. Efendiev, T. Y. Hou, and V. Ginting. “Multiscale finite element methods for nonlinear problems and their applications”. In: *Communications in Mathematical Sciences* 2.4 (2004), pp. 553–589.

[63] Y. Efendiev, T. Y. Hou, and X. Wu. “Convergence of a nonconforming multiscale finite element method”. In: *SIAM Journal on Numerical Analysis* 37.3 (2000), pp. 888–910.

[64] Y. Efendiev et al. “Generalized multiscale finite element methods. Oversamping strategies”. In: *International Journal for Multiscale Computational Engineering* 12.6 (2014), pp. 465–484.

[65] P. Emonot. “Méthodes de volumes éléments finis : applications aux équations de Navier-Stokes et résultats de convergence”. PhD thesis. Université Claude Bernard - Lyon I, 1992. URL: `http://www.theses.fr/1992LYO10280`.

[66] H. I. Ene. “Modeling the flow through porous media”. In: *Emerging Technologies and Techniques in Porous Media*. Springer, Netherlands, 2004, pp. 25–41.

[67] A. Ern and J.-L. Guermond. *Theory and practice of finite elements*. Springer-Verlag, New York, 2004.

[68] R. Ewing et al. “A smplified method for upscaling composite materials with high contrast of the conductivity”. In: *SIAM Journal on Scientific Computing* 31.4 (2009), pp. 2568–2586.

[69] R. Finn and D. R. Smith. “On the linearized hydrodynamical equations in two dimensions”. In: *Arch. Rational Mech. Anal.* 25 (1967), pp. 1–25.

[70] R. Finn and D. R. Smith. “On the stationary solutions of the Navier-Stokes equations in two dimensions”. In: *Arch. Rational Mech. Anal.* 25 (1967), pp. 26–39.

[71] A. Gerschenfeld, N. Forgione, and J. Thomas. “Multi-scale simulations of liquid metal systems”. In: *Thermal Hydraulics Aspects of Liquid Metal Cooled Nuclear Reactors*. 2019, pp. 361 –382.

[72] V. Ginting. “Analysis of two-scale finite volume element method for elliptic problem”. In: *Journal of Numerical Mathematics* 12.2 (2004), pp. 119–141.

[73] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations. Theory and algorithms*. Springer Series in Computational Mathematics. Springer-Verlag, Berlin, Heidelberg, 1986.

[74] V. Girault, V. Vassilev, and I. Yotov. “Mortar multiscale finite element methods for Stokes-Darcy flows”. In: *Numerische Mathematik* 127.1 (2014), pp. 93–165.

[75] I. G. Graham and R. Scheichl. “Robust domain decomposition algorithms for multiscale PDEs”. In: *Numer. Meth. Partial Diff. Eqns* 23.4 (2007), pp. 859–878.

[76] J.-L. Guermond. “Some implementations of projection methods for Navier-Stokes equations”. In: *Mathematical Modelling and Numerical Analysis* 30 (1996), pp. 637–667.

[77] J.-L. Guermond. “Un résultat de convergence d’ordre deux en temps pour l’approximation des équations de Navier-Stokes par une technique de projection incrémentale”. In: *Mathematical Modelling and Numerical Analysis* 33.1 (1999), pp. 169–189.

[78] J.-L. Guermond, P. Minev, and J. Shen. “An overview of projection methods for incompressible flows”. In: *Computer Methods in Applied Mechanics and Engineering* 195.44-47 (2006), pp. 6011–6045.

[79] J.-L. Guermond, P. Minev, and J. Shen. “Error analysis of pressure-correction schemes for the time-dependent Stokes equations with open boundary conditions”. In: *SIAM Journal on Numerical Analysis* 43.1 (2005), pp. 239–258.

[80] J.-L. Guermond and L. Quartapelle. “On stability and convergence of projection methods based on pressure Poisson equation”. In: *International Journal for Numerical Methods in Fluids* 26.9 (1998), pp. 1039–1053.

[81] F. Hecht. “New development in FreeFem++”. In: *Journal of Numerical Mathematics* 20.3-4 (2012), pp. 251–265.

[82] S. Heib. “Nouvelles discretisations non structurées pour des écoulements de fluides à incompressibilité renforcée”. PhD thesis. Université Paris VI, 2003. URL: `https://www.theses.fr/089439023`.

[83] P. Henning and M. Ohlberger. “The heterogeneous multiscale finite element method for elliptic homogenization problems in perforated domains”. In: *Numerische Mathematik* 113.4 (2009), pp. 601–629.

[84] P. Henning and D. Peterseim. “Oversampling for the multiscale finite element method”. In: *SIAM Multiscale Modeling & Simulation* 11.4 (2013), pp. 1149–1175.

[85] J. S. Hesthaven, S. Zhang, and X. Zhu. “High-order multiscale finite element methods for elliptic problems”. In: *Multiscale Modeling & Simulation* 12 (2014), pp. 650–666.

[86] U. Hornung. *Homogenization and porous media*. Springer-Verlag, New York, 1997.

[87] T. Y. Hou and X. H. Wu. “A multiscale finite element method for elliptic problems in composite materials and porous media”. In: *Journal of Computational Physics* 134.1 (1997), pp. 169–189.

[88] T. Y. Hou, X. H . Wu, and Z. Cai. “Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients”. In: *Mathematics of Computation* 68.227 (1999), pp. 913–943.

[89] T. Y. Hou, X. H. Wu, and Y. Zhang. “Removing the cell resonance error in the multiscale finite element method via a Petrov-Galerkin formulation”. In: 2.2 (2004), pp. 185–205.

[90] J. S. Howell and N. J. Walkington. “Inf-Sup conditions for twofold saddle point problems”. In: *Numer. Math.* 118.663 (2011), pp. 1–27.

[91] T. J. R. Hughes. “Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods”. In: *Computer Methods in Applied Mechanics and Engineering* 127.1-4 (1995), pp. 387–401.

[92] T. J. R. Hughes et al. “The variational multiscale method-a paradigm for computational mechanics”. In: *Comput. Methods Appl. Mech. Engrg.* 7825.98 (1998), pp. 3–24.

[93] O. Iliev, Z. Lakdawala, and V. Starikovicius. “On a numerical subgrid upscaling algorithm for Stokes-Brinkman equations”. In: *Computers and Mathematics with Applications* 65.3 (2013), pp. 435–448.

[94] O. Iliev, R. Lazarov, and J. Willems. “Fast numerical upscaling of heat equation for fibrous materials”. In: *Comput. Visual Sci.* 13.6 (2010), pp. 275–285.

[95] G. Jankowiak and A. Lozinski. “Non-conforming multiscale finite element method for Stokes flows in heterogeneous media. Part II: Error estimates for periodic microstrure”. In: *arXiv:1802.04389 [math.NA]* (2018).

[96] P. Jenny, S. H. Lee, and H. A. Tchelepi. “Multi-scale finite-volume method for elliptic problems in subsurface flow simulation”. In: *Journal of Computational Physics* 187.1 (2003), pp. 47–67.

[97] P. Jenny and I. Lunati. “Multi-scale finite volume method for elliptic problems with heterogeneous coefficients and source terms”. In: *Proceedings in Applied Mathematical Mechanics* 6.1 (2006), pp. 485–486.

[98] V. V. Jikov, S. M. Kozlov, and O. A. Oleinik. *Homogenizatino of differential operators and integral functions*. Springer-Verlag, Berlin, 1994.

[99] C. Johnson. *Numerical solutions of partial differential equations by the finite element method*. Cambridge University Press, New York, 1987.

[100] M. Kaviany. *Principles of heat transfer in porous media*. Springer-Verlag, New York, 1991.

[101] P. Lax and N. Milgram. *Parabolic equations. Contributions to the theory of partial differential equations. Annals of Mathematics Studies, no. 33.* Princeton University Press, Princeton, 1954, pp. 167–190.

[102] C. Le Bris, F. Legoll, and A. Lozinski. “An MsFEM type approach for perforated domains”. In: *Multiscale Modeling & Simulation* 12.3 (2014), pp. 1046–1077.

[103] C. Le Bris, F. Legoll, and A. Lozinski. “MsFEM à la Crouzeix-Raviart for highly oscillatory elliptic problems”. In: *Chinese Annuals of Mathematics. Series B* 34.1 (2012), pp. 113–138.

[104] C. Le Bris, F. Legoll, and F. Madiot. “A numerical comparison of some multiscale finite element approaches for advection-dominated problems in heterogeneous media”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 51.3 (2017), pp. 851–888.

[105] C. Le Bris, F. Legoll, and F. Madiot. “Multiscale finite element methods for advection-dominated problems in perforated domains”. In: *arXiv:1710.09331 [math.NA]* (2018).

[106] C. S. Lee. “Generalization of mixed multiscale finite element methods with applications”. PhD thesis. Texas A&M University, 2016. URL: `https://e-reports-ext.llnl.gov/pdf/825285.pdf`.

[107] C. S. Lee and D. Sheen. “Nonconforming generalized multiscale finite element methods”. In: *Journal of Computational and Applied Mathematics* 311 (2017), pp. 215–229.

[108] W. Li et al. “Preliminary study of coupling CFD code FLUENT and system code RELAP5”. In: *Annuals of Nuclear Energy* 73 (2014), pp. 96 –107.

[109] Y. Lin, J. Liu, and M. Yang. “Finite volume element methods: an overview on recent developments”. In: *International Journal of Numerical Analysis and Modeling, Series B* 4.1 (2013), pp. 14–34.

[110] X. Liu, J. Li, and Z. Chen. “A weak Galerkin finite element method for the Oseen equations”. In: *Adv Comput Math* 42.6 (2016), pp. 1473–1490.

[111] S. Lo. *Lecture CFD: CFD in nuclear thermal hydraulics*. URL: `https://www.imperial.ac.uk/media/imperial-college/research-centres-and-groups/nuclear-engineering/14-CFD-1.pdf`.

[112] S. P. Maclachlan and J. D. Moulton. “Multilevel upscaling through variational coarsening”. In: *Water Resources Research* 42.2005 (2006), pp. 1–9.

[113] F. Madiot. “Méthodes éléments finis de type MsFEM pour des problèmes d’advection diffusion”. PhD thesis. École des Ponts ParisTech, 2016. URL: `https://pastel.archives-ouvertes.fr/tel-01527285`.

[114] D. Martelli et al. “Coupled simulations of the NACIE facility using RELAP5 and ANSYS FLUENT codes”. In: *Annuals of Nuclear Energy* 101 (2017), pp. 408 –418.

[115] A. Moorthi, A.K. Sharma, and K. Velusamy. “A review of sub-channel thermal hydraulic codes for nuclear reactor core and future directions”. In: *Nuclear Engineering and Design* 332 (2018), pp. 329–344.

[116] J. D. Moulton, J. E. Dendy, and J. M. Hyman. “The black box multigrid numerical homogenization algorithm”. In: *Journal of Computational Physics* 142.1 (1998), pp. 80–108.

[117] B. P. Muljadi. “Multiscale method for Oseen problem in porous media with non-periodic grain patterns”. In: *Transport in Porous Media* 116.1 (2017), pp. 1–18.

[118] B. P. Muljadi et al. “Non-conforming multiscale finite element method for Stokes flows in heterogeneous media. Part I: methodologies and numerical experiments”. In: *Multiscale Model. Simul.* 13.4 (2015), pp. 1146–1172.

[119] Open CASCADE Technology. URL: `https://dev.opencascade.org/`.

[120] F. Ouaki. “Étude de schémas multi-échelles pour la simulation de réservoir”. PhD thesis. École Polytechnique, 2013. URL: `https://pastel.archives-ouvertes.fr/pastel-00922783/`.

[121] F. Ouaki et al. “A priori error estimate of a multiscale finite element method for transport modeling”. In: *SeMA J.* 67.1 (2014).

[122] PETSc. URL: `https://www.mcs.anl.gov/petsc/`.

[123] C. Philippe. “Développements et applications de méthodes numériques volumes finis pour la description d’écoulements océaniques”. PhD thesis. Université Joseph Fourier - Grenoble I, 1992. URL: `http://www.theses.fr/1996GRE10065`.

[124] D. Pialla et al. “Overview of the system alone and system/CFD coupled calculations of the PHENIX Natural Circulation Test within the THINS project”. In: *Nuclear Engineering and Design* 290 (2015), pp. 78–86.

[125] O. Pironneau. *Finite element method for fluids*. John Wiley & Sons, 1989.

[126] Pebble Bed Reactor. URL: https://en.wikipedia.org/wiki/Pebble-bed-reactor.

[127] RELAP5. URL: https://relap53d.inl.gov/SitePages/Home.aspx.

[128] D. S. Rowe. "Cross-Flow mixing between parallel flow channels during boiling Part-I COBRA-Computer program for coolant boiling in rod arrays". In: *BNWL-371-Pt-I* (1967).

[129] SALOME. URL: https://www.salome-platform.org/.

[130] E. Sanchez-Palencia. In: *Non-Homogeneous Media and Vibration Theory. Lecture Notes in Physics.* Vol. 127. Springer-Verlag, Berlin, Heidelberg, 1980, pp. 129–157.

[131] K. Simon and J. Behrens. "Multiscale finite elements through advection-induced coordinates for transient advection-diffusion equations". In: *arXiv:1802.07684 [math.NA]* (2018).

[132] D. R. Smith and R. Finn. "Estimates at infinity for stationary solutions of the Navier-Stokes equations in two dimensions". In: *Arch. Rational Mech. Anal.* 20 (1965), pp. 341–372.

[133] L. Tartar. *Incompressible fluid flow in a porous medium-convergence of the homogenization process, Appendix of [130].* 1980.

[134] R. Temam. *Navier-Stokes equations: theory and numerical analysis*. North–Holland publishing, Amsterdam - New York, 1979.

[135] R. Temam. "Une méthode d'approximation de la solution des équations de Navier-Stokes". In: *Bullutin de la S.M.F.* 96 (1968), pp. 115–152.

[136] F. Thomas. "Une méthode éléments finis à décomposition L2 d'ordre élevé motivée par la simulation d'écoulement diphasique bas Mach". PhD thesis. Université Paris VI, 2006. URL: https://www.theses.fr/2006PA066526.

[137] A. Toselli and O. Widlund. *Domain decomposition methods - Algorithms and theroy*. Springer-Verlag, Berlin Heidelberg, 2005.

[138] I. Toumi et al. "FLICA-4: a 3D two-phase flow computer code with advanced numerical methods for nuclear applications". In: *Nuclear Engineering and Design* 200 (2000), pp. 139–155.

[139] TrioCFD. URL: http://www-trio-u.cea.fr.

[140] P. S. Vassilevski. "Coarse spaces by algebraic multigrid: multigrid convergence and upscaling error estimates". In: *Advances in Adaptive Date Analysis* 3.1 & 2 (2011), pp. 229–249.

[141] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. 2nd ed. Pearson Education, 2007.

[142] VisIt. URL: https://wci.llnl.gov/simulation/computer-codes/visit/.

[143] M. Vohralik. "On the discrete Poincaré-Friedrichs inequalities for nonconforming approximations of the Sobolev space H1". In: *Numerical Functional Analysis and Optimization* 26.7-8 (2005), pp. 925–952.

[144] X. H. Wu, T. Y. Hou, and Y. Efendiev. "Analysis of upscaling absolute permeability". In: *Discrete and Continuous Dynamical Systems - Series B* 2.2 (2002), pp. 185–204.

[145] J. Xu and L. Zikatanov. "On an energy minimizing basis for algebraic multigrid methods". In: *Computing and Visualization in Science* 7.3-4 (2004), pp. 121–127.

[146] C. Zheng and H. Han. “Artificial boundary condition for exterior Oseen equation in 2-D space”. In: *Journal of Computational Mathematics* 20.6 (2002), pp. 591–598.

université
PARIS-SACLAY

École doctorale
de mathématiques
Hadamard (EDMH)

Titre : Développement d'une méthode d'éléments finis multi-échelles pour les écoulements incompressibles dans un milieu hétérogène

Mots clefs : élément de Crouzeix-Raviart, méthodes des éléments finis multi-échelles, équations de Navier-Stokes, équations de Stokes, milieu hétérogène

Résumé : Le cœur d'un réacteur nucléaire est un milieu très hétérogène encombré de nombreux obstacles solides et les phénomènes thermohydrauliques à l'échelle macroscopique sont directement impactés par les phénomènes locaux. Toutefois les ressources informatiques actuelles ne suffisent pas à effectuer des simulations numériques directes d'un cœur complet avec la précision souhaitée. Cette thèse est consacrée au développement de méthodes d'éléments finis multi-échelles (MsFEMs) pour simuler les écoulements incompressibles dans un milieu hétérogène avec un coût de calcul raisonnable. Les équations de Navier-Stokes sont approchées sur un maillage grossier par une méthode de Galerkin stabilisé, dans laquelle les fonctions de base sont solutions de problèmes locaux sur des maillages fins prenant précisément en compte la géométrie locale. Ces problèmes locaux sont définis par les équations de Stokes ou d'Oseen avec des conditions aux limites ou des termes sources appropriés. On propose plusieurs méthodes pour améliorer la précision des MsFEMs, en enrichissant l'espace des fonctions de base locales. Notamment, on propose des MsFEMs d'ordre élevée dans lesquelles ces conditions aux limites et termes sources sont choisis dans des espaces de polynômes dont on peut faire varier le degré. Les simulations numériques montrent que les MsFEMs d'ordre élevés améliorent significativement la précision de la solution. Une chaîne de simulation multi-échelle est construite pour simuler des écoulements dans des milieux hétérogènes de dimension deux et trois.

Title: Development of a multiscale finite element method for incompressible flows in heterogeneous media

Keys words : Crouzeix-Raviart element, multiscale finite element method, Navier-Stokes equations, Stokes equations, heterogeneous media

Abstract: The nuclear reactor core is a highly heterogeneous medium crowded with numerous solid obstacles and macroscopic thermohydraulic phenomena are directly affected by localized phenomena. However, modern computing resources are not powerful enough to carry out direct numerical simulations of the full core with the desired accuracy. This thesis is devoted to the development of Multiscale Finite Element Methods (MsFEMs) to simulate incompressible flows in heterogeneous media with reasonable computational costs. Navier-Stokes equations are approximated on the coarse mesh by a stabilized Galerkin method, where basis functions are solutions of local problems on fine meshes by taking precisely local geometries into account. Local problems are defined by Stokes or Oseen equations with appropriate boundary conditions and source terms. We propose several methods to improve the accuracy of MsFEMs, by enriching the approximation space of basis functions. In particular, we propose high-order MsFEMs where boundary conditions and source terms are chosen in spaces of polynomials whose degrees can vary. Numerical simulations show that high-order MsFEMs improve significantly the accuracy of the solution. A multiscale simulation chain is constructed to simulate successfully flows in two- and three-dimensional heterogeneous media.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France