ADAPTIVE AND PARALLEL VARIATIONAL MULTISCALE METHOD FOR THE NAVIER-STOKES EQUATIONS

XIE CONG

Ph.D

The Hong Kong Polytechnic University

2015

The Hong Kong Polytechnic University

Department of Applied Mathematics

Adaptive and parallel variational multiscale method for the Navier-Stokes equations

Cong Xie

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

April 2015

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

Cong Xie (Name of student)

Dedicate to my family and my parents.

Abstract

Navier-Stokes equations are basic equations in fluid dynamic. The problem is important both in practice and theory. As it is difficult to find their accuracy solutions, numerical simulations and experimentations have become important approaches to solve the problem. Variational multiscale finite element method is one of most useful methods. In order to guarantee the effectiveness, adaptive algorithm has been developed, which makes use of the solutions in the progress to automatically control the computing progress. In this thesis we first present an adaptive variational multiscale method for the Stokes equations. Then we develop two kinds of variational multiscale method based on the partition of unity for the Navier-Stokes equations.

First, we propose some a posterior error indicators for the variational multiscale method for the Stokes equations and prove the equivalence between the indicators and the error of the finite element discretization. Some numerical experiments are presented to show their efficiency on constructing adaptive meshes and controlling the error.

Secondly, a parallel variational multiscale method based on the partition of unity is proposed for incompressible flows. Based on two-grid method, this algorithm localizes the global residual problem of variational multiscale method into a series of local linearized residual problems. To decrease the undesirable effect of the artificial homogeneous Dirichlet boundary condition of local sub-problems, an oversampling technique is also introduced. The globally continuous finite element solutions are constructed by assembling all local solutions together using the partition of unity functions. Especially, we add an artificial stabilization term in the local and parallel procedure by considering the residual as a subgrid value, which keeps the sub-problems stable. We present the theoretical analysis of the method and numerical simulations demonstrate the high efficiency and flexibility of the new algorithm.

Another a partition of unity parallel variational multiscale method is proposed. The main difference lies in that in this algorithm we propose two kinds of refinement method. It is difficult to obtain the theoretical result as the above method. However, the numerical simulations show that the error of this algorithm decays exponentially with respect to the oversampling parameter.

Acknowledgements

I am grateful to the several individuals who have supported me in various ways during my PhD program and would like to hereby acknowledge their assistance.

First and foremost, I wish to express my deep thanks to my supervisor, Porf. Yanping Lin, not only for his enlightening guidance, but also for his continuous encouragement during all my study period. I am deeply grateful for his consideration and support.

I would like to give my heartfelt thanks to Dr. Haibiao Zheng for his helpful suggestions on my research work.

I want to give my biggest thanks to my family. Thanks for my parents who always love me unconditionally. And thanks my husband Xianfeng Zeng's for his patiently waiting and supporting through times of difficulty. Especially I want to thank my baby girl Yixiao.

Last but not least, I would like to express gratitude to all the colleagues and our department AMA.

Contents

List of Figures

List of Tables

List of Notations

$\mathbb{R}$	set of real numbers
$\mathbb{C}$	set of complex numbers
$\mathbb{R}^d$	set of d-dimensional real vectors
Ω	bounded domain in R^d
$\partial\Omega$	boundary of Ω
$H^k(\Omega)$	Sobolev space defined on Ω
Re	Reynolds number
$\mathbf{u}$	velocity vector
p	pressure
$\mathbf{f}$	prescribed body force
ν	kinematic viscosity

Chapter 1

Introduction

1.1 Background

The variational multiscale method was proposed to solve multiscale problems by Hughes and co-workers in [42, 44]. They defined a projection of the large scales in Large Eddy Simulation method into appropriate subspaces. Since then much attention has been paid in this field. For example, John and Kaya [46] gave the finite element analysis of a variational multiscale method for the Navier-Stokes equations. Gravemeier et al. [33] also presented the three-level variational multiscale method. Zheng et al. improved the finite element variational multiscale method by introducing two Gauss integration method [89] and adaptive technique [90]. Zhang et al. [88], Yu et al. [86], Shan et al. [64] presented subgrid model, projection basis and modular type to improve the variational multiscale methods, respectively.

Based on the observation that in numerical simulations low frequency components can be approximated well by the relative coarse grid and high frequency components can be computed on a fine grid by some local and parallel procedure, the parallel finite element computations have been widely used [83, 12, 36, 71]. Combining the partition of unity method [59, 6] and the parallel adaptive algorithm from [12], Holst [38, 39] constructed the parallel partition of unity method (PPUM). Zheng et al. [87, 91] developed some local and parallel finite element algorithms based on the partition of unity. Song et al. [73] presented an adaptive local postprocessing technique based on the partition of unity method for the Navier-Stokes equations. There are also some papers improving the variational multiscale methods by combining with two-grid method or local and parallel techniques [52, 66].

It is natural to consider to add the local parallel method to the variational multiscale method in order to retain the best features of both methods and overcome many of their defects. In particular, we use the variational multiscle method based on two local Gauss integrations [89] since it avoids constructing the projection operator, keeps the same efficiency and does not need extra storage compared with common VMS method. Comparing with the parallel method in [66], we add an artificial stabilization term in the local and parallel procedure by considering the residual as a subgrid value, which keeps the sub-problems stable. Then, an oversampling

technique is introduced in order to overcome the undesirable effect of the artificial homogeneous Dirichlet boundary conditions of local sub-problems. The interesting points in this algorithm lie in: firstly, a class of partition of unity is derived by a given triangulation, which guides the domain decomposition; secondly, the series of local linearized residual problems are implemented in parallel, and they require less communication between each other; finally, the globally continuous finite element solution is obtained by assembling all local solutions together via the partition of unity functions.

1.2 Literature review

1.2.1 Navier-Stokes equations

The Navier-Stokes equations describe the motion of fluid substances in physics. They are named after Claude-Louis Navier(1785-1863) and George Gabriel Stokes(1981-1903). Assuming that the fluid stress is the sum of a diffusing viscous term and a pressure term, we can derive these equations by applying Newton's second law to fluid motion. Exactly, we can get the equations from the conservation of mass and conservation of momentum [57].

Because the equations describe the physics of many things of academic and economic interest they have a lot of applications. In many fields, such as petroleum

industry, plasma physics, the Navier-Stokes equations appear alone or coupled with other equations. For example coupled with Maxwell's equations they can be used to model and study magnetohydrodynamics.

In a purely mathematical sense the Navier-Stokes equations are also of great interest. However, although people have studied the equations since the 19th century mathematicians have not yet proven the existence and smoothness of the solutions of the equations in three dimensions, which are called as existence and smoothness problems of Navier-Stokes equations. Since May 2000 the Clay Mathematics Institute has offered a 1,000,000 dollar prize for people who can make substantial progress in mathematical theory to help people understand the equations, which has been called this one of the seven most important open problems in mathematic.

It is difficult to find the accurate solutions of the Navier-Stokes equations. People have found only more than one hundred particular solutions. That's one of the reasons that numerical simulations are introduced to solve the problem. The finite difference method, finite volume method and finite element method are the three most commonly used methods to solve partial differential equations.

1.2.2 Variational multiscale method

Variational multiscale method proposed by Hughes [42, 44] offers a new perspective for the stabilized methods. They defined the large scales in Large Eddy Simulation method a projection into appropriate subspaces. In the method, they decomposed the solution into fine scales and coarse scales $\mathbf{u} = \mathbf{u}_c + \mathbf{u}_f$, determined the fine scale solution $\mathbf{u}_f$ analytically, eliminated it from the coarse equation and then solved the coarse scale solution $\mathbf{u}_c$ numerically. Based on the idea [42], [43], people proposed several variational multiscale methods [44], [45], [47]. In [47], John and Kaya gave the finite element analysis of a variational multiscale method for the Navier-Stokes equations. In [54], Liu et al. presented a new variational multiscale method for the Stokes problem by using the infinite Green's function to derive the fine scale velocity and pressure analytically.

Gravemeier et al. [33] also presented the three-level variational multiscale method. Zheng et al. improved the finite element variational multiscale method by introducing two Gauss integration method [89] and adaptive technique [90]. Zhang et al. [88], Yu et al. [86], Shan et al. [64] et al. presented subgrid model, projection basis and modular type to improve the variational multiscale methods, respectively.

1.2.3 Adaptive technique

With the development of computers, numerical computation plays a more and more important role in our research and practice. Not only can they simulate a lot of physical phenomenon, but also can prove the reliability of theoretical analysis. However, no matter how complex the mathematical model is, there still exists numerical error which could influence the effectiveness of the method. In order to guarantee the effectiveness, scientist build mathematical theory to estimate the discretization error. Furthermore, the discretization error also provides us with important information to use the adaptive algorithm to get better numerical solutions.

Adaptive algorithm uses the solutions in the progress to automatically control the computing progress to solve the problems. It can first compute the mesh by the solutions in the progress and then choose the best discretization pattern, thus it can adapt the error to the needed accuracy step by step. The method can use less amount of calculation to achieve the needed accuracy. Adaptive analysis consists of two parts: reliable error estimate and powerful automatical mesh division technology. Thus the a posteriori error analysis is the essential part in the adaptive algorithm.

The a posteriori error estimates for finite element methods was first proposed in the pioneering work of Babuška and Rheinboldt in 1978 [8] when they presented a posteriori error estimates of finite element solutions of linear elliptic problems in

one dimension. Then some other results were presented [9, 10, 7]. These results formed the foundation of adaptive algorithm. In the early 1980s, based on prior error estimate or difference value people derived lots of a posteriori error estimators, which were still rough but very efficient for the adaptive algorithm. Zienkiewicz and Zhu proposed a simple error estimation technology which could be used for the finite element method for many classical problems [63], [92]. In 1990s the basic technic of a posteriori error estimation was built and had been used to the classical problems, such as Stokes equations, Navier-Stokes equations, hyperbolic problems and parabolic problems. Ainsworth and Oden [2],[3] Stewart and Hughes [74, 75] and many other scientists did much work in this area. Now the theory of a posteriori error estimation has been mature and the key point of the research has turned to test the effectiveness of the a posteriori error estimators and their limitations in computing.

Especially for the Stokes and Navier-Stokes equations, R.Verfürth gave a posteriori error analysis of the finite element method for Stokes equations [80]. The main contribution was that he proposed a posteriori error estimator for the mini-element discretization of the Stokes equations and proved the equivalence between the estimator and the discretization error by using the property of Bubble function. Then the method was applied to non-confirming element discretization of Stokes equations

and Navier-Stokes equations [81, 82],.

Larson and Målqvist developed two adaptive variational multiscale method for elliptic problem in [50, 51]. They decoupled and solved the fine scale equations approximately on patches, derived a posteriori error estimate to control a linear functional of the error or got a posteriori error estimate of the error in the energy norm. Then they got the adaptive algorithms based on the a posteriori error estimators. In [35] they derived an explicit a posteriori error estimator using the approximation of the Green's function. John and Kindl developed a method which could choose the large scale space adaptively in [48]. It extended the projection-based variational multiscale finite element method. In [90] the authors gave an adaptive variational multiscale method in which the error estimators could be computed by two local Gauss integrations at the element method.

1.2.4 The partition of unity method

In the beginning people only used the polynomial basis functions in the trial space of the standard FEM. The idea of adding non-polynomial basis functions into the trail space of the FEM started in the 1970s [15, 16, 28].

Duarte and Oden [25, 26, 53, 61, 24] developed a new meshless method called h-p clouds, the basic idea of which is to multiply a partition of unity by polynomials

or other appropriate class of functions. Both a priori and a posterior error estimate were analyzed and the implementation of the method using objective oriented programming was discussed.

In [5] Babuška and co-authors proposed three Special FEM to solve second order problems with rough coefficients. In particular, the shape functions used in the Special FEM ♯3 have compact supports and are products of piecewise linear FE hat-functions and a non-polynomial function that mimic the special features of the unknown solution. Melenk [58] further generalized the method with detailed mathematical theory and applications in his Ph.D dissertation. He also showed that the hat-functions could be replaced by any PU (with compact support).

Then Babuška and Melenk referred to the method as the partition of unity finite element method(PUFEM) in [59, 6]. They explained the mathematical foundation of the PUFEM and discussed some of its features, for example, the ability to include a priori knowledge about the local behavior of the solution in the finite element space, the ability to construct finite element spaces of any desired regularity. They also showed how to use the PUFEM to employ the sructure of the differential equation under consideration to construct effective and robust methods. Some classes of non-standard problems which can profit highly from the advantages of the PUFEM were presented.

A lot of attention has been paid on the method to the method. Two approaches are the Generalized Finite Element Method(GFEM) which are proposed by Stouboulis [76, 77, 78] and Duarte [61, 24, 27] and the Extended Finite Element Method(XFEM), which are raised by Belytschko and co-atuthors [13, 60, 21, 14, 22]. Recently it was recognized that these two methods are same and were referred to as XFEM/GFEM [30].

Huang and Xu [40] first applied a partition of unity method to homogenization problems. They made use of the advantage that the partition of unity method could flexibly localize the approximation and keep the global continuity and showed that the partition of unity method was a powerful tool for handling a large variation of problems efficiently. Then they [41] proposed a finite element method for nonmatching overlapping grids considering both overlapping and nonoverlapping cases.

Bank and Holst [11, 12] presented the parallel partition of unity method (PPUM) by combining the local and parallel method and the partition of unity method. Then Holst described a variant of the algorithm in [38, 39]. Global error estimates for PPUM were derived and a duality-based variant of PPUM was also discussed.

1.2.5 Local and parallel finite element algorithm

The finite element method was proposed by R. Courant in 1943 [20]. However, people didn't notice the contribution of the method until the engineers independently re-invented the method in the 1950s [4, 79]. R. Clough proposed the name in 1960 [18]. It has become one of the major tools for numerically solving partial differential equations based on the advantages that it can relatively easily handle general boundary conditions, complex geometry, and variable material properties. Also, it is possible for people to develop flexible and general purpose software for application based on the clear structure and versatility of the method. Moreover, a solid theoretical foundation makes it possible to obtain concrete error estimates of the finite element solutions in many situations.

When we use finite element method to solve the Navier-Stokes equations we will meet two difficult problems [23]. First, due to the discretization of the nonlinear convective terms we have to use stabilized finite element formulations to treat high Reynolds number flows. Secondly, it is difficult to numerically handle the saddle-point problem will arise from the variational form of the incompressible problem with the pressure acting as a Lagrangian multiplier of the incompressibility constraint. A lot of work have been done to overcome these difficulties, such as, the penalty method [42], pressure gradient method [19], projection method, and so on.

Xu and Zhou prosposed some new local and parallel discretization and adaptive finite element algorithms for elliptic boundary value problems in [83]. They use a coarse grid to capture the global component of the approximate solution and then parallelize the major computation in a much fine grid. The local error estimate for finite element approximations is one important technical tool in this idea. Then they applied the similar method to nonlinear elliptic boundary value problems in both two and three dimensions [84]. The method is also useful for the eigenvalue problems [85, 72]. Bank and Holst [11, 12] presented a similar approach and implemented in detail.

Based on two-grid discretizations, He et al. [37, 36] developed some new local and parallel finite element algorithms for the Stokes problems and the stationary incompressible Navier-Stokes problem. They also obtained some local a prior error estimates for the finite element solutions on general shape-regular grids. Ma et al. applied the method to the stream function form of Navier-Stokes equations [56] and steady Navier-Stokes equations [55].

Shang et al. have published many papers in this topic. They proposed and analyzed the method for the d-dimensional (d=2, 3) transient Stokes equations [69]. Several different kinds of local and parallel finite element algorithms for the Navier-Stokes equations were presented in succession [67, 71, 70, 65]. Based on two-grid

discretizations and domain decomposition, a parallel Oseen-linearized finite element algorithm for the stationary Navier-Stokes equations with moderate or large viscosity parameter was proposed and analyzed in [68]. They also proposed a new method by combining the two-grid discretization approach with a finite element variational multiscale algorithm for simulation of the incompressible Navier-Stokes equations [66].

1.3 Summary of contributions of the thesis

The original contributions of this thesis are as follows:

- We propose some a posterior error indicators for the variational multiscale method for the Stokes equations and prove the equivalence between the indicators and the error of the finite element discretization. Some numerical experiments are presented to show their efficiency on constructing adaptive meshes and controlling the error.

- A parallel variational multiscale method based on the partition of unity is proposed for incompressible flows. Based on two-grid method, this algorithm localizes the global residual problem of variational multiscale method into a series of local linearized residual problems. To decrease the undesirable effect of the artificial homogeneous Dirichlet boundary condition of local sub-problems,

an oversampling technique is also introduced. The globally continuous finite element solutions are constructed by assembling all local solutions together using the partition of unity functions. Especially, we add an artificial stabilization term in the local and parallel procedure by considering the residual as a subgrid value, which keeps the sub-problems stable. We present the theoretical analysis of the method and numerical simulations demonstrate the high efficiency and flexibility of the new algorithm.

Another a partition of unity parallel variational multiscale method is proposed. The main difference lies in that in this algorithm we propose two kinds of refinement method. It is difficult to obtain the theoretical result as the above method. However, the numerical simulations show that the error of this algorithm decays exponentially with respect to the oversampling parameter.

1.4 Organization of the thesis

The thesis is structured as follows.

- Chap. 2 reviews the preliminary knowledge of Navier-Stokes and Stokes equations. Some notations and properties are introduced.

- Chap. 3 presents some a posterior error indicators for the variational multiscale method for the Stokes equations and proves the equivalence between the

indicators and the error of the finite element discretization. Some numerical experiments are presented to show their efficiency on constructing adaptive meshes and controlling the error.

- Chap. 4 proposes two kinds of parallel variational multiscale methods based on the partition of unity for incompressible flows. Based on two-grid method, the algorithms localizes the global residual problem of variational multiscale method into a series of local linearized residual problems.

  In the first method, to decrease the undesirable effect of the artificial homogeneous Dirichlet boundary condition of local sub-problems, an oversampling technique is also introduced. The globally continuous finite element solutions are constructed by assembling all local solutions together using the partition of unity functions. Especially, we add an artificial stabilization term in the local and parallel procedure by considering the residual as a subgrid value, which keeps the sub-problems stable. We present the theoretical analysis of the method and numerical simulations demonstrate the high efficiency and flexibility of the new algorithm.

  Another a partition of unity parallel variational multiscale method is then proposed. The main difference lies in that in this algorithm we propose two kinds of refinement methods. It is difficult to obtain the theoretical result as the

above method. However, the numerical simulations show that the error of this algorithm decays exponentially with respect to the oversampling parameter.

- Chap. 5 concludes the whole thesis and lists our plans in the future work.

Chapter 2

Preliminaries

2.1 Navier-Stokes equations

We consider the following incompressible flows

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f} \quad \texttt{in } \Omega, \\ \nabla\cdot\mathbf{u} &= 0 \quad \texttt{in } \Omega, \\ \mathbf{u} &= 0 \quad \texttt{on } \partial\Omega, \end{aligned} \tag{2.1}$$

where Ω represents a polyhedral domain in R^d (d =2, 3) with boundary $\partial\Omega$, $\mathbf{u}$, p, $\mathbf{f}$ and $\nu > 0$ represent the velocity vector, pressure, prescribed body force, kinematic viscosity respectively. And ν is inversely proportional to the Reynolds number Re.

For sub-domains $D \subset G \subset \Omega$, $D \subset\subset G$ means that $\mathrm{dist}(\partial D\backslash\partial\Omega, \partial G\backslash\partial\Omega) > 0$. Throughout the thesis we use C to denote a generic positive constant whose value

may change from place to place but remains independent of the mesh parameter h.

In order to describe the variational form of the equations we give some notations. For a bounded domain $\Omega \subset R^d$, we use the standard notations for Sobolev spaces $W^{s,k}(\Omega)$ and their associated norms [1, 17]. Especially when $k = 2$, $H^s(\Omega) = W^{s,2}(\Omega)$ denotes the usual Soblev space, $\|\cdot\|_{s,\Omega} = \|\cdot\|_{s,2,\Omega}$ denotes standard Soblev norm, $(\cdot,\cdot)_s$ denotes the inner product in $L^2(\Omega)$ or its vector value version. We introduce the following spaces,

$$\begin{aligned} H_0^1(\Omega) &= \{\varphi \in H^1(\Omega); \varphi = 0 \quad \text{on} \quad \Gamma\}, \\ L_0^2(\Omega) &= \{\varphi \in L^2(\Omega); \int_\Omega \varphi = 0\}. \end{aligned}$$

Let $X = (H_0^1(\Omega))^2$, $Y = L^2(\Omega)^2$, $M = L_0^2(\Omega)$. $H^{-1}(\Omega)$ is the dual space of $H_0^1(\Omega)$. In the following we will denote the spaces consisting of vector-valued functions in boldface.

Then we define the bilinear terms $a(\cdot,\cdot)$, $d(\cdot,\cdot)$ and trilinear term $b(\cdot,\cdot,\cdot)$ as follows

$$a(\mathbf{u},\mathbf{v}) = \nu(\nabla\mathbf{u}, \nabla\mathbf{v}),$$

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) = \tfrac{1}{2}((\mathbf{u}\cdot\nabla)\mathbf{v},\mathbf{w}) - \tfrac{1}{2}((\mathbf{u}\cdot\nabla)\mathbf{w},\mathbf{v}),$$

$$d(\mathbf{v},q) = (\nabla\cdot\mathbf{v},q), \forall\ \mathbf{u},\mathbf{v},\mathbf{w} \in \mathbf{X}, q \in M,$$

With the above notations, the standard variational formulation of (2.1) reads: find $(\mathbf{u}, p) \in (\mathbf{X}, M)$ such that

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{u}, \mathbf{u}, \mathbf{v}) - d(\mathbf{v}, p) + d(\mathbf{u}, q) = (\mathbf{f}, \mathbf{v}), \forall\ (\mathbf{v}, q) \in (\mathbf{X}, M). \tag{2.2}$$

Here are some properties of $a(\mathbf{u}, \mathbf{v})$, $b(\mathbf{u}, \mathbf{v}, \mathbf{w})$, $d(\mathbf{v}, p)$.

(1) $a(\cdot, \cdot)$ is symmetric, continuous and X coercive.

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= a(\mathbf{v}, \mathbf{u}), \quad \forall \mathbf{u}, \mathbf{v} \in X; \\ |a(\mathbf{u}, \mathbf{v})| &\leqslant \lambda |\mathbf{u}|_1 |\mathbf{v}|_1, \quad \forall \mathbf{u}, \mathbf{v} \in X; \\ a(\mathbf{u}, \mathbf{u}) &= \lambda |\mathbf{u}|_1^2, \quad \forall \mathbf{u} \in X. \end{aligned}$$

(2) $b(\cdot, \cdot, \cdot)$ has the following properties:

$$b(\mathbf{u}, \mathbf{v}, \mathbf{w}) = b(\mathbf{u}, \mathbf{w}, \mathbf{v}), \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in X;$$

$$|b(\mathbf{u}, \mathbf{v}, \mathbf{w})| \leqslant N \|\mathbf{u}\| \|\mathbf{v}\| \|\mathbf{w}\|, \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in X;$$

$$|b(\mathbf{u}, \mathbf{v}, \mathbf{w})| \leqslant C(\|\mathbf{u}\| \|\mathbf{v}\|^{\frac{1}{2}} |\mathbf{v}|^{\frac{1}{2}} + \|\mathbf{v}\| \|\mathbf{u}\|^{\frac{1}{2}} |\mathbf{u}|^{\frac{1}{2}}) \|\mathbf{w}\|^{\frac{1}{2}} |\mathbf{w}|^{\frac{1}{2}}, \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in X;$$

$$|b(\mathbf{u}, \mathbf{v}, \mathbf{w})| \leqslant C(\|\mathbf{v}\| \|\mathbf{w}\|^{\frac{1}{2}} |\mathbf{w}|^{\frac{1}{2}} + \|\mathbf{w}\| \|\mathbf{v}\|^{\frac{1}{2}} |\mathbf{v}|^{\frac{1}{2}}) \|\mathbf{u}\|^{\frac{1}{2}} |\mathbf{u}|^{\frac{1}{2}}, \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in X.$$

(3) $d(\cdot,\cdot)$ is continuous and satisfy the inf-sup condition.

$$\sup_{\mathbf{v}\in X}\frac{d(\mathbf{v},q)}{|\mathbf{v}|_1}\geqslant\beta\|q\|_0,\quad\forall q\in M$$

Then we give the following classicla existence and uniqueness of solution of (2.2), see [32, 34].

Theorem 2.1. *Given* $\mathbf{f}\in\mathbf{X}'$*, there exists at least a solution pair* $(\mathbf{u},p)\in(\mathbf{X},M)$ *which satisfies (2.2) and*

$$||\nabla\mathbf{u}||_{0,\Omega}\leqslant\nu^{-1}||\mathbf{f}||_{-1,\Omega},||\mathbf{f}||_{-1,\Omega}=\sup_{\mathbf{v}\in\mathbf{X}}\frac{(\mathbf{f},\mathbf{v})}{||\nabla\mathbf{v}||_{0,\Omega}},$$

and if ν *and* $\mathbf{f}$ *satisfy the following uniqueness condition:*

$$1-\nu^{-2}N||\mathbf{f}||_{-1,\Omega}\geqslant 0, \tag{2.3}$$

then the solution pair $(\mathbf{u},p)$ *of problem (2.2) is unique.*

2.2 Variational multiscale method for Navier-Stokes equations

Let τ_h be a regular triangulation of the domain Ω, and h denote the maximum diameter of the elements in τ_h. We use P_2-P_1 elements in this paper, which

means that $\mathbf{X}^h$ and M^h contain piecewise polynomials of degree 2 and 1 respectively. $(\mathbf{X}^H, M^H)$ is defined in the same way but on τ_H with coarser mesh size H where $H > h$. Set $(\mathbf{X}_0^h, M_0^h) = (\mathbf{X}^h, M^h) \bigcap (\mathbf{X}, M)$.

It is known that the standard Galerkin finite element discretization of (2.2) is unstable in the case of high Reynolds number (or smaller viscosity). Therefore, we consider the finite element variational multiscale method [89]: find $(\mathbf{u}_h, p_h) \in (\mathbf{X}_0^h, M_0^h)$ satisfying

$$\nu a(\mathbf{u}_h, \mathbf{v}) + b(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}) - d(\mathbf{v}, p_h) + d(\mathbf{u}_h, q) + G(\mathbf{u}_h, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall (\mathbf{v}, q) \in (\mathbf{X}_0^h, M_0^h), \tag{2.4}$$

where $G(\mathbf{u}_h, \mathbf{v}) = \alpha((I - \Pi_h)\nabla \mathbf{u}_h, (I - \Pi_h)\nabla \mathbf{v})$. Let $\mathbb{L} = L^2(\Omega)^{d\times d}$ and $\Pi_h : \mathbb{L} \to \mathbb{L}^h$ be the orthogonal projection operator with the following properties:

$$((I - \Pi_h)\mathbf{r}, \mathbf{g}_h) = 0, \quad \forall \mathbf{r} \in \mathbb{L}, \ \mathbf{g_h} \in \mathbb{L}^{\mathbf{h}}, \tag{2.5}$$

$$\|\Pi \mathbf{r}\|_0 \leqslant C\|\mathbf{r}\|_0, \quad \forall \mathbf{r} \in \mathbb{L}, \tag{2.6}$$

$$\|(I - \Pi_h)\mathbf{r}\|_0 \leqslant Ch\|\mathbf{r}\|_1, \quad \forall \mathbf{r} \in \mathbb{L} \cap H^1(\Omega)^{d\times d}, \tag{2.7}$$

where I is the identify operator.

According to [89], we can use the equivalent formulation of G based on two local

Gauss integrations as follows,

$$G(\mathbf{u}_h, \mathbf{v}) = \alpha \sum_{\Omega_e \in \tau_h} \{ \int_{\Omega_e, s} \nabla \mathbf{u}_h \nabla \mathbf{v} \mathbf{d}x - \int_{\Omega_e, 1} \nabla \mathbf{u}_h \nabla \mathbf{v} \mathbf{d}x \} \quad \forall \mathbf{u}_h,\ \mathbf{v} \in \mathbf{X}^h,$$

where $\int_{\Omega_e, i} g(x)\mathbf{d}x$ denotes an appropriate Gauss integral over Ω_e which is exact for polynomials of degree i, $i = 1, 2$. For all test functions $\mathbf{v} \in \mathbf{X}^h$, $\nabla \mathbf{u}_h$ must be piecewise constant when $i = 1$. And set $\alpha = O(h^2)$ in order to keep the rates of convergence.

In [66] Shang proved the following theorem.

Theorem 2.2. *Assume that* $(\mathbf{u}, p)$ *is a nonsingular solution to the Navier-Stokes equations satisfying* $(\mathbf{u}, p) \in (H^3(\Omega)^n \bigcap H_0^1(\Omega)^n) \times (H^2(\Omega) \bigcap L_0^2(\Omega))$, *and* α *tends to zero as* h *tends to zero. Then the solution* $(\mathbf{u}_h, p_h)$ *computed by the numerical scheme (2.4) satisfies*

$$\|\mathbf{u} - \mathbf{u_h}\|_{\mathbf{1},\mathbf{\Omega}} + \|\mathbf{p} - \mathbf{p_h}\|_{\mathbf{0},\mathbf{\Omega}} \leqslant \mathbf{ch^2} + \mathbf{c}\alpha, \tag{2.8}$$

$$\|\mathbf{u} - \mathbf{u_h}\|_{\mathbf{0},\mathbf{\Omega}} + \|\mathbf{p} - \mathbf{p_h}\|_{\mathbf{-1},\mathbf{\Omega}} \leqslant \mathbf{ch}(\mathbf{h^2} + \alpha) + \mathbf{c}\alpha^{\mathbf{2}}. \tag{2.9}$$

Chapter 3

Adaptive variational multiscale method for the Stokes equations

3.1 Variational multiscale finite element method for Stokes equations

We give the variational form of Stokes equations: find $(\mathbf{u}, p) \in V \times M$ satisfied

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, p) &= (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in V, \\ d(\mathbf{u}, q) &= 0, \quad \forall q \in M. \end{cases} \tag{3.1}$$

Define $X = V \times M$, then the variational form is equivalent to the following equality:

$$\mathcal{L}(\mathbf{u}, p; \mathbf{v}, q) = a(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, p) + d(\mathbf{u}, q) = (\mathbf{f}, \mathbf{v}), \quad \forall (\mathbf{v}, q) \in X, \tag{3.2}$$

In the following we present the variational multiscale formulation of the problem. First we split the space X into two subspaces: the coarse space X_c and the fine space X_f which satisfy:

$$X = X_c \oplus X_f = V_c \times M_c \oplus V_f \times M_f,$$

where $V = V_c \oplus V_f$ and $M = M_c \oplus M_f$. Thus we can get the following decomposition

$$\mathbf{u} = \mathbf{u}_c \oplus \mathbf{u}_f, \mathbf{v} = \mathbf{v}_c \oplus \mathbf{v}_f,$$

$$p = p_c \oplus p_f, q = q_c \oplus q_f.$$

Then the problem can be divided into two sub problems as follows,

$$\mathcal{L}(\mathbf{u}_c, p_c; \mathbf{v}_c, q_c) + \mathcal{L}(\mathbf{u}_f, p_f; \mathbf{v}_c, q_c) = (\mathbf{f}, \mathbf{v}_c), \quad \forall (\mathbf{v_c}, q_c) \in X_c, \tag{3.3}$$

$$\mathcal{L}(\mathbf{u}_c, p_c; \mathbf{v}_f, q_f) + \mathcal{L}(\mathbf{u}_f, p_f; \mathbf{v}_f, q_f) = (\mathbf{f}, \mathbf{v}_f), \quad \forall (\mathbf{v_f}, q_f) \in X_f, \tag{3.4}$$

We define the residual $R : X \to X_f$ as follows

$$R((\mathbf{v}, q), (\mathbf{w}, t)) = (\mathbf{f}, \mathbf{w}) - \mathcal{L}(\mathbf{v}, q; \mathbf{w}, t), \quad \forall (\mathbf{w}, t) \in X.$$

Thus the fine scale solution can be driven by the residual of the coarse scale solution

as follows,

$$\mathcal{L}(\mathbf{u}_f, p_f; \mathbf{v}_f, q_f) = R((\mathbf{u}_c, p_c), (\mathbf{v}_f, q_f)), \quad \forall (\mathbf{v}_f, q_f) \in X_f. \tag{3.5}$$

Denote $\mathbf{u}_f = F_1 R(\mathbf{u}_c, p_c)$, $q_f = F_2 R(\mathbf{u}_c, p_c)$ then we can derive the modified coarse scale equation,

$$\mathcal{L}(\mathbf{u}_c, p_c; \mathbf{v}_c, q_c) + \mathcal{L}(F_1 R(\mathbf{u}_c, p_c), F_2 R(\mathbf{u}_c, p_c); \mathbf{v}_c, q_c) = (\mathbf{f}, \mathbf{v}_c), \quad \forall (\mathbf{v_c}, q_c) \in X_c. \tag{3.6}$$

Here the second term is the effect of the fine scales on the coarse scales.

In the following, we present the localized problems to solve the fine scale equation. First we introduce the triangulations of Ω: $\mathcal{J}_h$ which satisfies

- the intersection of two different elements is at most a vertex or a whole edge,
- the ratio of the diameter of any element in $\mathcal{J}_h$ to the diameter of its inscribed circle is bounded by a constant independent of h.

Let T present the element in $\mathcal{J}_h$ with diameter h_T and N be the set of nodes in $\mathcal{J}_h$. Now we present how to choose the local domain. Denote $\{\phi_i\}_{i \in N}$ to be a set of Lagrange basis function in X_c. We know that $\{\phi_i\}_{i \in N}$ is a partition of unity with support on the elements sharing the nodes i. Based on this we make choice of the local domain. Let S_i^1 be the set of the elements which have one corner in node i

and $S_i^k = \sum_{j \in S_i^{k-1}} S_1^j$ and we call it the kth layer of node i. We choose the local domain ω_i to be $S_i^k (k = 1, 2, 3 \cdots)$. In Fig 3.1 we present $S_i^k (k = 1, 2, 3)$ and the fine h-refinement meshes for them.

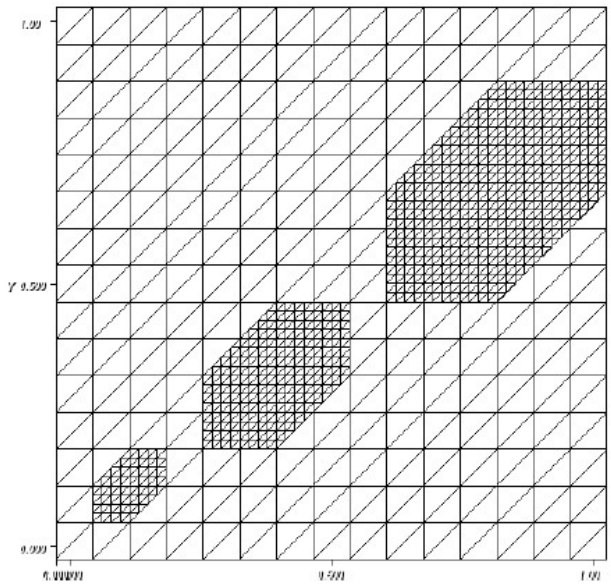

Figure 3.1: 1st layer(left), 2nd layer, and 3rd layer(right)

Thus we can approximate the fine scale equation by constructing an algorithm using a set of decoupled localized problems. The fine scale solution can be written as $\mathbf{u}_f = \sum_{i \in N} \mathbf{u}_{f,i}$, $p_f = \sum_{i \in N} p_{f,i}$, where

$$\mathcal{L}(\mathbf{u}_{f,i}, p_{f,i}; \mathbf{v}_f, q_f) = \phi_i R((\mathbf{u}_c, p_c), (\mathbf{v}_f, q_f)), \quad \forall (\mathbf{v}_f, q_f) \in X_f. \tag{3.7}$$

We introduce this into (3.4), then the Stokes problem becomes: find $(\mathbf{u}_c, p_c) \in X_c$

and $\mathbf{u}_f = \sum_{i\in N} \mathbf{u}_{f,i} \in V_f$, $p_f = \sum_{i\in N} p_{f,i} \in M_f$ such that

$$\mathcal{L}(\mathbf{u}_c, p_c; \mathbf{v}_c, q_c) + \mathcal{L}(\mathbf{u}_f, p_f; \mathbf{v}_c, q_c) = (\mathbf{f}, \mathbf{v}_c), \quad \forall(\mathbf{v_c}, q_c) \in X_c, \tag{3.8}$$

$$\mathcal{L}(\mathbf{u}_{f,i}, p_{f,i}; \mathbf{v}_f, q_f) = \phi_i R((\mathbf{u}_c, p_c), (\mathbf{v}_f, q_f)), \quad \forall(\mathbf{v}_f, q_f) \in X_f. \tag{3.9}$$

Then we use finite element method to solve this problem. First we choose the finite element spaces X_H and X_h to approximate the coarse space X_c and X_f respectively. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ denote the space of linear polynomials and quadratic polynomials. Put

$$\begin{aligned} P_1 &= \{\varphi \in C(\overline{\Omega}) : \forall T \in \mathcal{J}_H, \varphi|_T \in \mathcal{P}_1\}, \\ P_2 &= \{\varphi \in C(\overline{\Omega}) : \forall T \in \mathcal{J}_H, \varphi|_T \in \mathcal{P}_2\}, \\ V_H &= \{P_2 \cap H_0^1(\Omega)\}^2, \quad M_H = P_1 \cap L_0^2(\Omega). \end{aligned}$$

There are several choices of the fine scale space. In our computing, we choose to use standard piecewise polynomials to discrete X_f on the fine mesh as that of the coarse mesh.

Then the discretization form of the problem is: find $(\mathbf{U}_c, P_c) \in X_H$ and $\mathbf{U}_f =$

$\sum_{i\in N} \mathbf{U}_{f,i} \in V_h$, $P_f = \sum_{i\in N} P_{f,i} \in M_h$ such that

$$\mathcal{L}(\mathbf{U}_c, P_c; \mathbf{v}_c, q_c) + \mathcal{L}(\mathbf{U}_f, P_f; \mathbf{v}_c, q_c) = (\mathbf{f}, \mathbf{v}_c), \forall (\mathbf{v_c}, q_c) \in X_H, \tag{3.10}$$

$$\mathcal{L}(\mathbf{U}_{f,i}, p_{f,i}; \mathbf{v}_f, q_f) = \phi_i R((\mathbf{U}_c, P_c), (\mathbf{v}_f, q_f)), \forall (\mathbf{v}_f, q_f) \in X_h(\omega_i) \text{and} i \in N. \tag{3.11}$$

Obviously, $\mathbf{U} = \mathbf{U}_c + \mathbf{U}_f$ and $P = P_c + P_f$ are continuous because in $X_h(\omega_i)$ the functions are equal to zero on the boundary $\partial\omega_i$. In our computing, we only choose some nodes to solve the local problems. Thus we denote F as a set of the nodes where the local problems need to be solved and C as the set of other nodes

3.2 A posteriori error analysis for variational multiscale FEM for Stokes equations

In this section we define two kinds of a posteriori error estimators to estimate the discretization error. The first kind is based on a quasi-dual problem and the second one computes suitable norms of the residual.

Denote $\mathbf{e} = \mathbf{u} - \mathbf{U}$ and $\varepsilon = p - P$ to be the error. It's obvious that the errors can be divided into two parts: the coarse scale error $\mathbf{e}_c = \mathbf{u}_c - \mathbf{U}_c$, $\varepsilon_c = p_c - P_c$ and the fine scale error $\mathbf{e}_f = \sum_{i\in N} \mathbf{e}_{f,i} = \sum_{i\in N}(\mathbf{u}_{f,i} - \mathbf{U}_{f,i})$, $\varepsilon_f = \sum_{i\in N} \varepsilon_{f,i} = \sum_{i\in N}(p_{f,i} - P_{f,i})$.

Now we derive some properties of the error. Subtracting (3.10) from (3.3) we can

get the Galerkin orthogonality of the coarse scale error

$$\mathcal{L}(\mathbf{e}_c, \varepsilon_c; \mathbf{v}_c, q_c) + \mathcal{L}(\mathbf{e}_f, \varepsilon_f; \mathbf{v}_c, q_c) = 0, \quad \forall(\mathbf{v_c}, q_c) \in X_H. \tag{3.12}$$

And the Galerkin orthogonality of the fine scale error can be derived by subtracting (3.11) from (3.4):

$$\mathcal{L}(\mathbf{e}_{f,i}, \varepsilon_f; \mathbf{v}_f, q_f) = -\mathcal{L}(\mathbf{e}_c, \varepsilon_c; \phi_i \mathbf{v}_f, \phi_i q_f), \quad \forall(\mathbf{v_f}, q_f) \in X_h. \tag{3.13}$$

We first define for any $T \in \mathcal{J}_h$,

$$\eta_{C,T} = \{ch_T \|P_0\mathbf{f} - \Delta\mathbf{U}_c - \nabla P_c\|_{0,T}^2 + ch_E^{\frac{1}{2}} \|[\frac{\partial \mathbf{U}_c}{\partial n} - P_c \cdot n]_J\|_{0,E}^2 + \|\nabla \cdot \mathbf{u}_h\|_{0,T}^2\}^{\frac{1}{2}} \tag{3.14}$$

and for any $T \in F$,

$$\eta_{F,T} = \{ch_T \|\Delta\mathbf{U}_c + \nabla P_c\|_{0,T}^2 + ch_E^{\frac{1}{2}} \|[\frac{\partial \mathbf{U}_c}{\partial n} - P_c \cdot n]_J\|_{0,E}^2 + \|\nabla \cdot \mathbf{u}_h\|_{0,T}^2\}^{\frac{1}{2}} \tag{3.15}$$

where the $[\cdot]_J$ denotes the jump of $(\cdot)$ across E.

Theorem 3.1. *The a posteriori error estimate*

$$\begin{aligned} |\mathbf{e}|_1 + \|\varepsilon\|_0 \quad &\leqslant \quad c\{\{\sum_{T\in C}(\eta_{C,T}^2 + h_T^2\|\mathbf{f} - P_0\mathbf{f}\|_{0,T}^2)\}^{\frac{1}{2}} \\ &\quad +\{\sum_{T\in F}(\eta_{C,T}^2 + h_T^2\|\mathbf{f} - P_0\mathbf{f}\|_{0,T}^2 + \eta_{F,T}^2)\}^{\frac{1}{2}}\} \end{aligned} \tag{3.16}$$

holds, where c only depends on Ω *and the smallest angle in the triangulation* $\mathcal{J}_h$.

Proof. It's known that the P2-P1 velocity-pressure finite element space $\overline{X} = X_H \oplus X_h$ satisfies the inf-sup condition, that is there exist $\beta > 0$ such that

$$\inf_{(\mathbf{u},p)\in\overline{X}} \sup_{(\mathbf{v},q)\in\overline{X}} \frac{\mathcal{L}(\mathbf{u},p;\mathbf{v},q)}{(|\mathbf{u}|_1 + \|p\|_0)(|\mathbf{v}|_1 + \|q\|_0)} \geqslant \beta.$$

In the proof we have to use the following two inequalities[80] for all $T \in \mathcal{J}_h$ and all edges E of T,

$$\|\mathbf{v} - I_h\mathbf{v}\|_{0,T} \leqslant Ch_T|\mathbf{v}|_{1,T},$$

and

$$\|\mathbf{v} - I_h\mathbf{v}\|_{0,E} \leqslant Ch_T^{\frac{1}{2}}|\mathbf{v}|_{1,E}.$$

Here $I_h\mathbf{v}$ is the standard pointwise interpolation operator by the finite elements.

$$
\begin{aligned}
& \mathcal{L}(\mathbf{e},\varepsilon;\mathbf{v},q) \\
= \;& \mathcal{L}(\mathbf{e},\varepsilon;\mathbf{v}_f - I_h\mathbf{v}_f, q_f) \\
= \;& \mathcal{L}(\mathbf{u}-U_c, p-P_c;\mathbf{v}_f - I_h\mathbf{v}_f, q_f) - \mathcal{L}(\mathbf{U}_f, P_f;\mathbf{v}_f - I_h\mathbf{v}_f, q_f) \\
= \;& \sum_{i\in C}\{(\mathbf{f}-\Delta\mathbf{U}_c-\nabla P_c, \mathbf{v}_f - I_h\mathbf{v}_f)_T + (q_f, \nabla\cdot\mathbf{U}_c)_T + \frac{1}{2}\sum_{E\subset\partial T\cap\Omega}([\frac{\partial\mathbf{U}_c}{\partial n} - P_c\cdot n]_J, \\
& \mathbf{v}_f - I_h\mathbf{v}_f)_E\} + \sum_{i\in F}\{(\mathbf{f}-\Delta\mathbf{U}_c-\nabla P_c, \mathbf{v}_f - I_h\mathbf{v}_f)_T + (q_f, \nabla\cdot\mathbf{U}_c)_T \\
& +\frac{1}{2}\sum_{E\subset\partial T\cap\Omega}([\frac{\partial\mathbf{U}_c}{\partial n} - P_c\cdot n]_J, \mathbf{v}_f - I_h\mathbf{v}_f)_E - ((-\Delta\mathbf{U}_f - \nabla P_f, \mathbf{v}_f - I_h\mathbf{v}_f)_T \\
& +(q_f, \nabla\cdot\mathbf{U}_f)_T + \frac{1}{2}\sum_{E\subset\partial T\cap\Omega}([\frac{\partial\mathbf{U}_f}{\partial n} - P_f\cdot n]_J, \mathbf{v}_f - I_h\mathbf{v}_f)_E)\} \\
\leqslant \;& \sum_{T\in C}\{ch_T\|P_0\mathbf{f}-\Delta\mathbf{U}_c-\nabla P_c\|_{0,T}|\mathbf{v}_f|_1 + ch_E^{\frac{1}{2}}\|[\frac{\partial\mathbf{U}_c}{\partial n} - P_c\cdot n]_J\|_{0,E}|\mathbf{v}_f|_1 \\
& +ch_T\|\mathbf{f}-P_0\mathbf{f}\|_{0,T}|\mathbf{v}_f|_1 + \|\nabla\cdot\mathbf{u}_h\|_{0,T}\|q_f\|_{0,T}\} + \sum_{T\in F}\{ch_T\|P_0\mathbf{f}-\Delta\mathbf{U}_c \\
& -\nabla P_c\|_{0,T}|\mathbf{v}_f|_1 + ch_E^{\frac{1}{2}}\|[\frac{\partial\mathbf{U}_c}{\partial n} - P_c\cdot n]_J\|_{0,E}|\mathbf{v}_f|_1 + ch_T\|\mathbf{f}-P_0\mathbf{f}\|_{0,T}|\mathbf{v}_f|_1 \\
& +\|\nabla\cdot\mathbf{U}_c\|_{0,T}\|q_f\|_{0,T} + (ch_T\|\Delta\mathbf{U}_f + \nabla P_f\|_{0,T}|\mathbf{v}_f|_1 + ch_E^{\frac{1}{2}}\|[\frac{\partial\mathbf{U}_f}{\partial n} \\
& -P_f\cdot n]_J\|_{0,E}|\mathbf{v}_f|_1 + \|\nabla\cdot\mathbf{U}_f\|_{0,T}\|q_f\|_{0,T})\} \\
\leqslant \;& c\{\{\sum_{T\in C}(\eta_{C,T}^2 + h_T^2\|\mathbf{f}-P_0\mathbf{f}\|_{0,T}^2)\}^{\frac{1}{2}} + \{\sum_{T\in F}(\eta_{C,T}^2 + h_T^2\|\mathbf{f}-P_0\mathbf{f}\|_{0,T}^2 \\
& +\eta_{F,T}^2)\}\}^{\frac{1}{2}}\{|\mathbf{v}|_{1,T} + \|q\|_{0,T}\}.
\end{aligned}
$$

Thus we can get the conclusion.□

As the adaptive algorithm based on the second error estimator is very similar to that based on the first one, we also denote $C_i = \eta_{C,T}$ and $F_i = (\eta_{C,T}^2 + \eta_{F,T}^2)^{\frac{1}{2}}$.

3.3 Numerical experiments

An adaptive algorithm usually has four steps:

$$\text{solve} \rightarrow \text{estimate} \rightarrow \text{remark} \rightarrow \text{refine}$$

The algorithms based on the two kinds of a posteriori error estimator are similar, thus we present an adaptive algorithm as follows.

- Step 1. Give an initial mesh $\mathcal{J}_0$ with no node in F.
- Step 2. Solve the Stokes problem (3.1) on the mesh $\mathcal{J}_i$.
- Step 3. Calculate C_i for each coarse node.
- Step 4. Solve the local problems where C_i is large and then we can get the new U_f and U_c.
- Step 5. Calculate C_i and F_i. If C_i is large, solve the local problems; if F_i is large, we choose to increase the number of layers for the local problem. Based

on C_i and F_i generate mesh $\mathcal{J}_{i+1}$ by the refinement strategy in [29]. Stop if we get the desired tolerance or go back to Step 2.

Then we present two examples to show the efficiency of our error estimators in the process of constructing self-adaptive meshes and in estimating the discretization errors.

Example 3.1 We consider a driven cavity problem in the domain $\Omega = (0,1) \times (0,1)$. It means that $u_x = 1$, $u_y = 0$ on the upper side and $\mathbf{u} = \mathbf{0}$ on the other three sides. Fig 3.2 presents the problem. It's obvious that the solution is not continuous on the two vertices of the upper boundary.

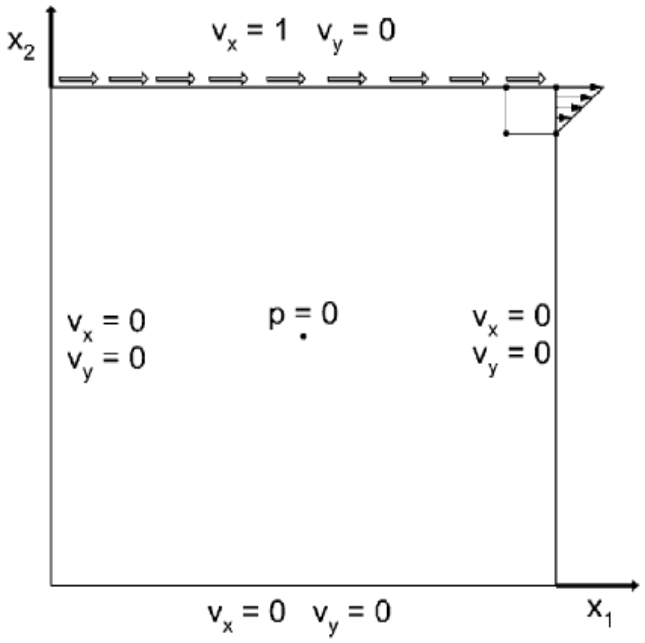

Figure 3.2: Driven cavity problem

We begin our calculation from the initial mesh in Fig 3.3, Fig 3.4 and Fig 3.5

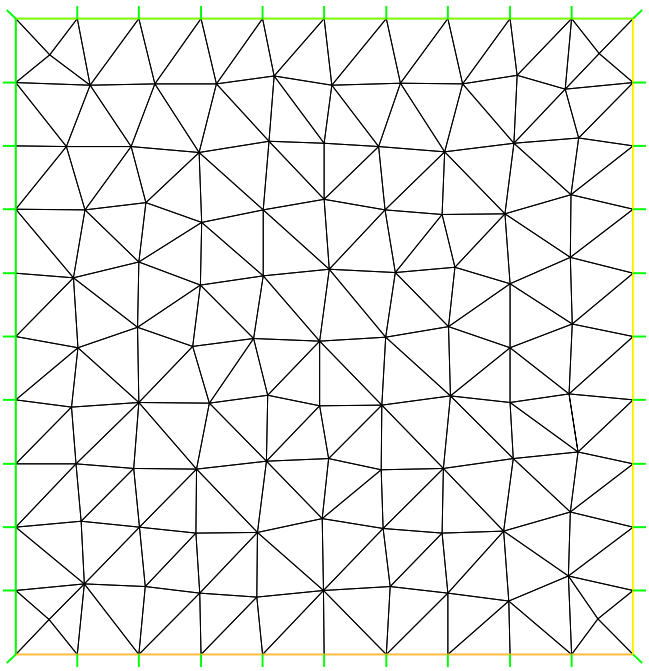

Figure 3.3: initial mesh

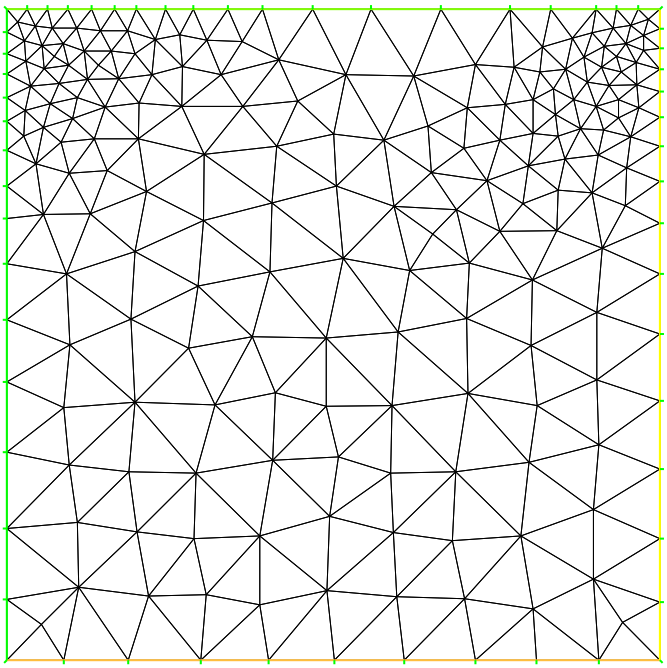

Figure 3.4: mesh after 1 refinement

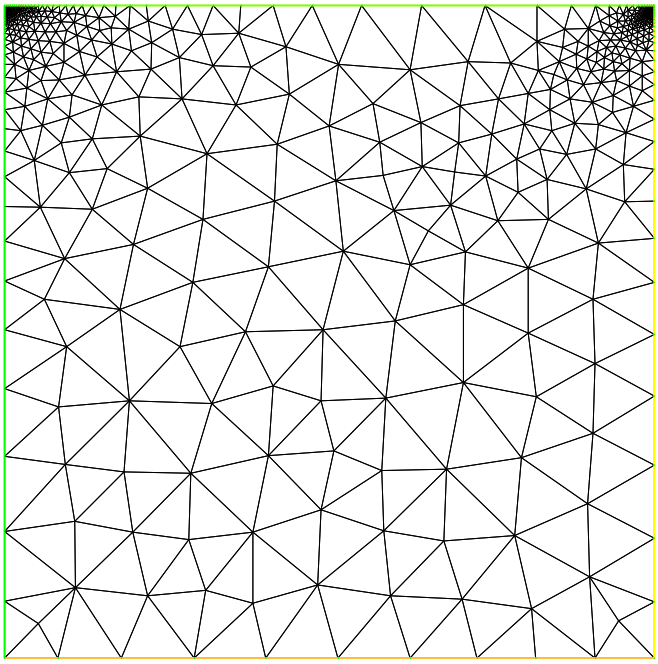

Figure 3.5: mesh after 3 refinements

present the mesh we get after one refinement and three refinements respectively. From these pictures we can see clearly that at the two top corners there are more triangles than other areas. In Fig 3.8 and Fig 3.9 we present the velocity field in uniform mesh and adaptive form mesh after 2 steps in Fig 3.6 and Fig 3.7 with nearly the same number of triangles. From these figures we can see that the solution using the a posteriori error analysis is more closer to the real situation.

Example 3.2 We consider a problem with a smooth solutions which are given

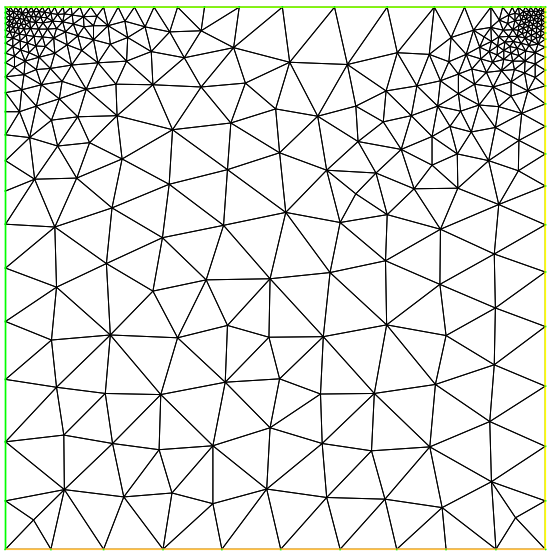

Figure 3.6: mesh after 2 refinements

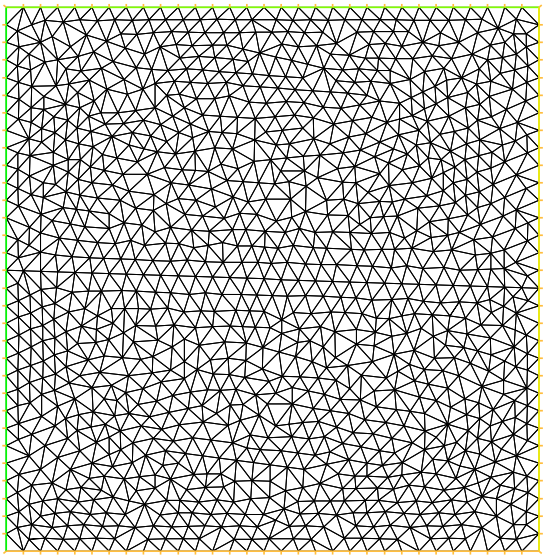

Figure 3.7: uniform mesh

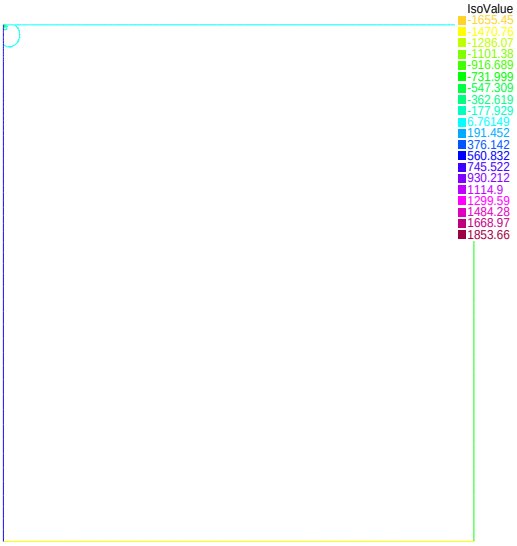

Figure 3.8: P on mesh after 2 refinements

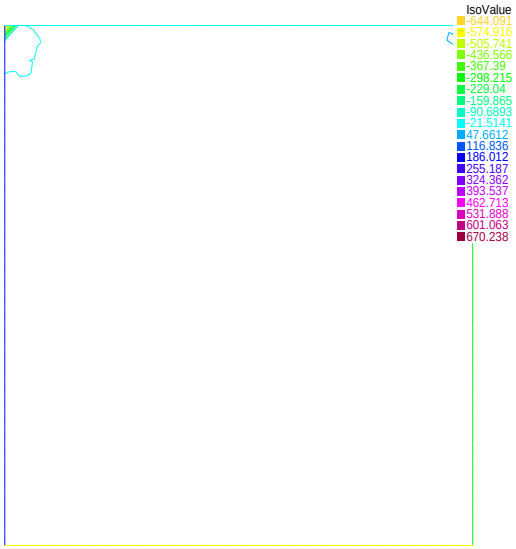

Figure 3.9: P on uniform mesh

by

$$u_x = 1.5r^{1/2}(cos(0.5\theta) - cos(1.5\theta)),$$

$$u_y = 1.5r^{1/2}(3sin(0.5\theta) - sin(1.5\theta)),$$

$$p = -6/r^{1/2}cos(0.5\theta),$$

in a circular domain with radius 1 and angle 2π and with a non-homogeneous Dirichlet boundary conditions on the curved part of the boundary and homogeneous Dirichlet boundary conditions on the straight part of the boundary. We start the strategies from the initial triangulations $\mathcal{J}_h^0$ as in Fig 3.10 and refine 2 times to get the meshes $\mathcal{J}_h^1$ and $\mathcal{J}_h^2$ shown in the following two figures. We can see that in the noncontinuous area there are much more elements than these in the continuous area. Fig 3.13 -Fig 3.18 present the pressure level line with Galerkin method and VMS method on the above three meshes. It can be seen clearly that the pressure based on VMS method is more smooth. From the result we can obviously see the advantage of our method.

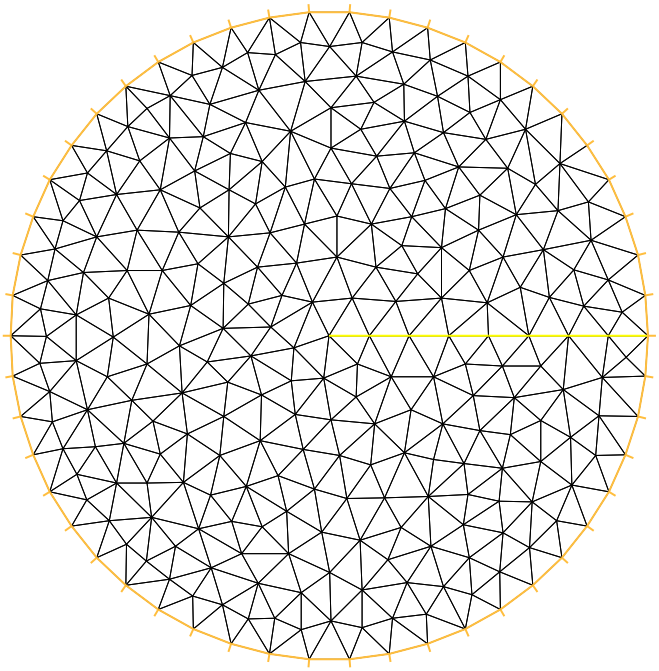

Figure 3.10: initial mesh

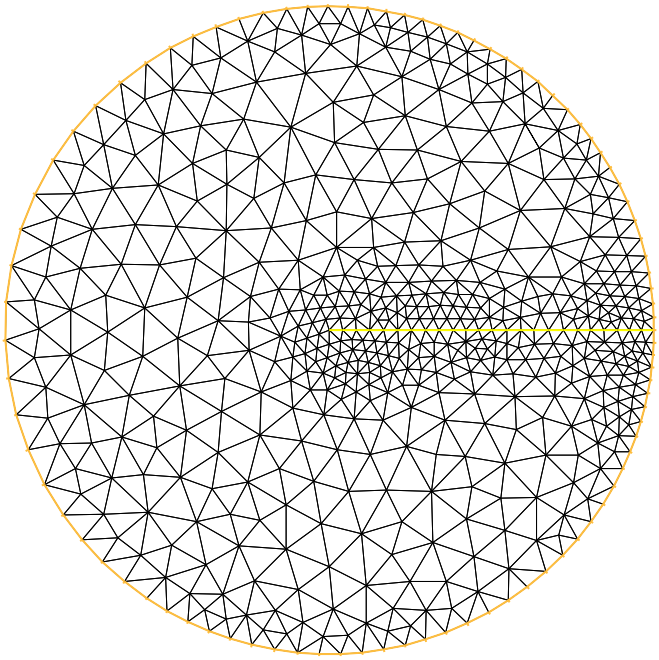

Figure 3.11: mesh after 1 refinement

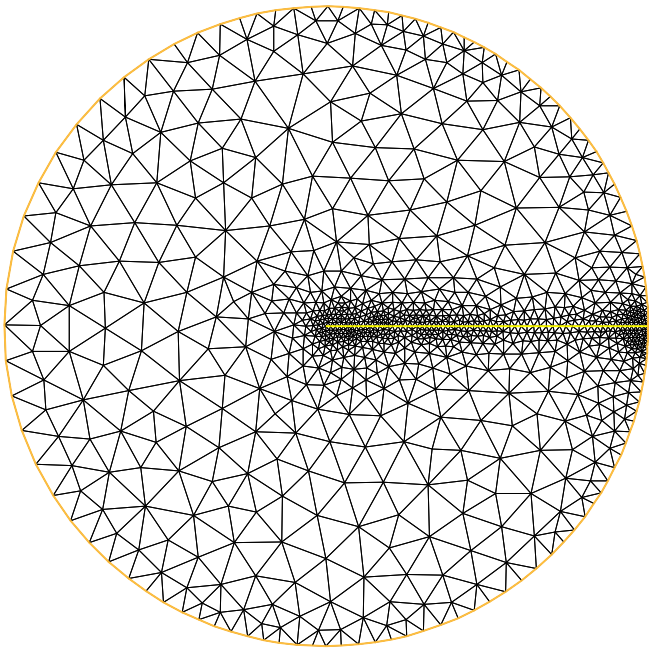

Figure 3.12: mesh after 2 refinements

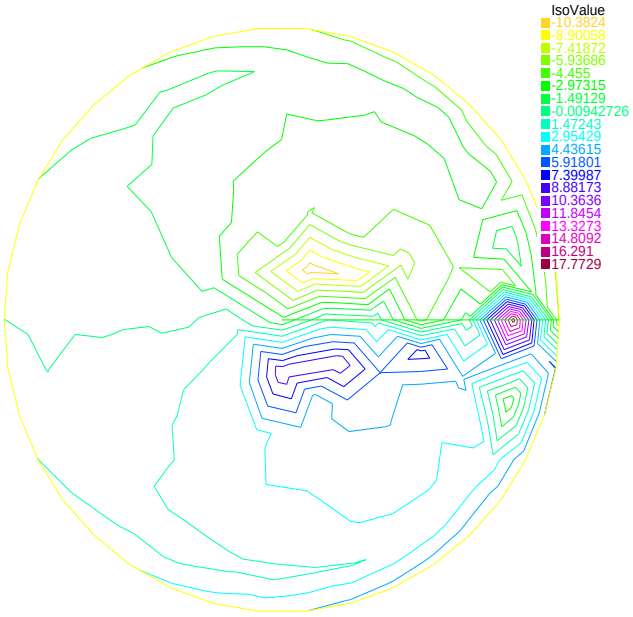

Figure 3.13: pressure level line with Galerkin method on $\mathcal{J}_h^0$

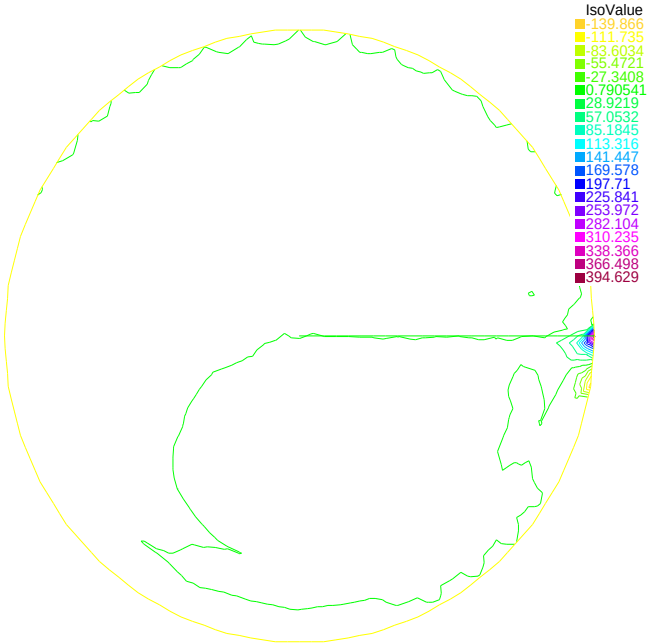

Figure 3.14: pressure level line with VMS method on $\mathcal{J}_h^0$

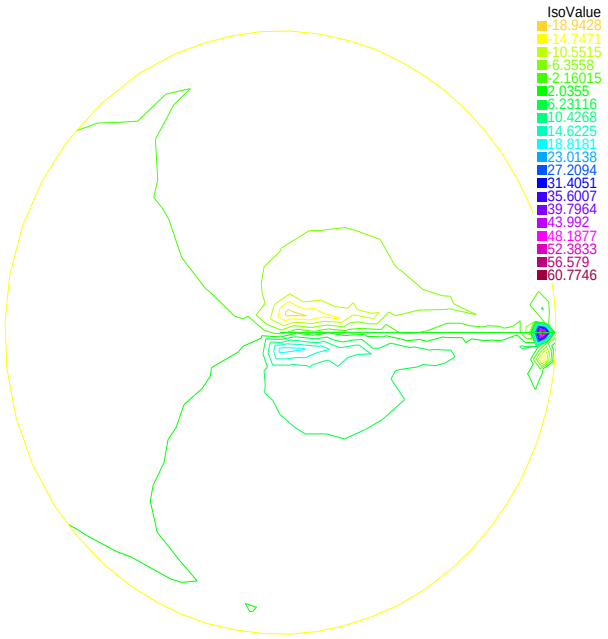

Figure 3.15: pressure level line with Galerkin method on $\mathcal{J}_h^1$

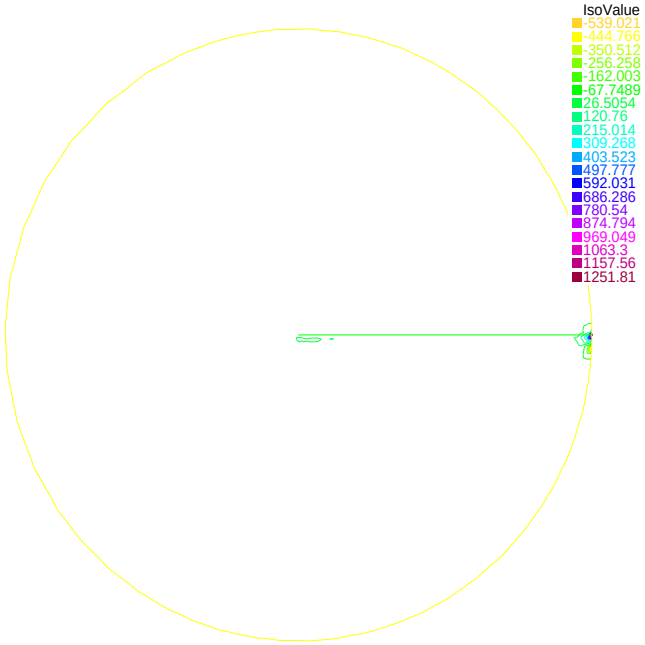

Figure 3.16: pressure level line with VMS method on $\mathcal{J}_h^1$

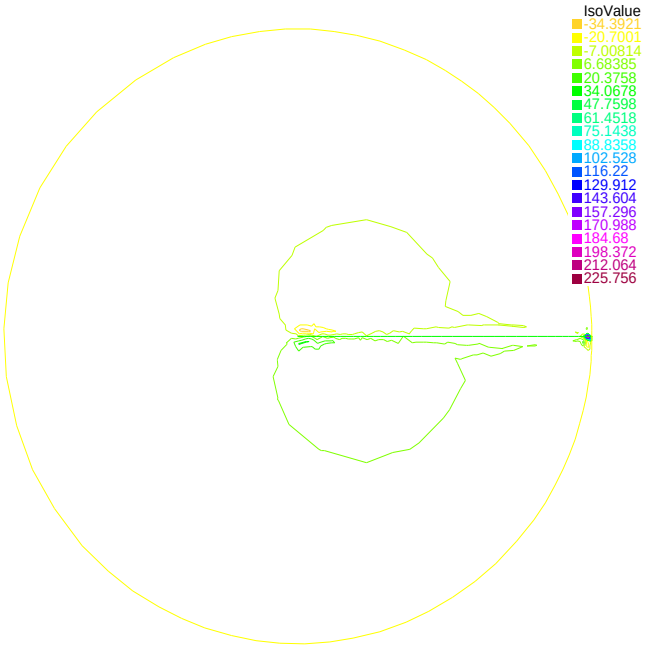

Figure 3.17: pressure level line with Galerkin method on $\mathcal{J}_h^2$

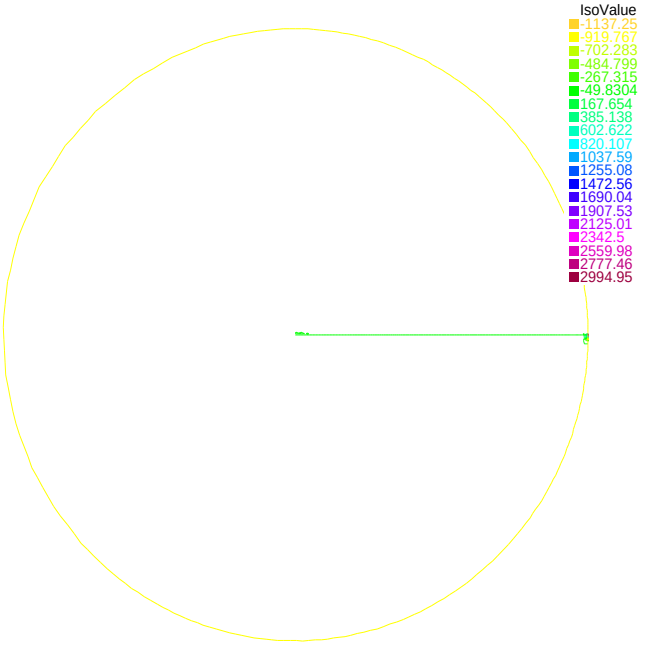

Figure 3.18: pressure level line with VMS method on $\mathcal{J}_h^2$

Chapter 4

A parallel variational multiscale methods for Navier-Sotkes equations based on the partition of unity

4.1 The parallel variational multiscale method based on the partition of unity

4.1.1 Analysis and algorithm

In this section, we will derive a partition of unity based on a given triangulation, and propose a framework for domain decomposition.

First, choose a regular conforming triangulation τ_{H_p} for Ω. For each node $x_i \in$

τ_{H_p}, $i = 1, 2, \cdots, N$, (here N is the number of nodes on τ_{H_p}), define associate continuous linear Lagrange basis function φ_i, such that $\varphi_i(x_m) = \delta_{i,m}$. Let $\omega^i = supp\varphi_i \cap \Omega$, $i = 1, 2, \cdots, N$ denote the local subdomain.

Then, we denote $\omega^{i,0} = \omega^i$, which means the local domain without oversampling. To enlarge this domain we introduce one layer oversampling $\omega^{i,1}$, which is the union of the supports of φ_i and one layer of its neighbors, and also multiple layers oversampling $\omega^{i,s}$:

$$\omega^{i,1} = \bigcup_{x_m \in \omega^{i,0}} \omega^m,$$

$$\omega^{i,s} = \bigcup_{x^m \in \omega^{i,s-1}} \omega^m.$$

Fig4.1 would help us to understand the definition of oversampling.

It's easy to check that, for any given s, $\{\omega^{i,s}\}_1^N$ is an open cover of Ω and $\{\varphi_i\}_1^N$ is a partition of unity subordinate to the cover $\{\omega^{i,s}\}_1^N$ which satisfying

$$\texttt{supp}\varphi_i \subset \overline{\omega^{i,s}}, \ \forall i. \tag{4.1}$$

$$\sum_i \varphi_i \equiv 1 \ \texttt{on} \ \Omega. \tag{4.2}$$

$$||\varphi_i||_{L^\infty(R^n)} \leqslant C_\infty. \tag{4.3}$$

$$||\nabla\varphi_i||_{L^\infty(R^n)} \leqslant \frac{C_G}{H_p}. \tag{4.4}$$

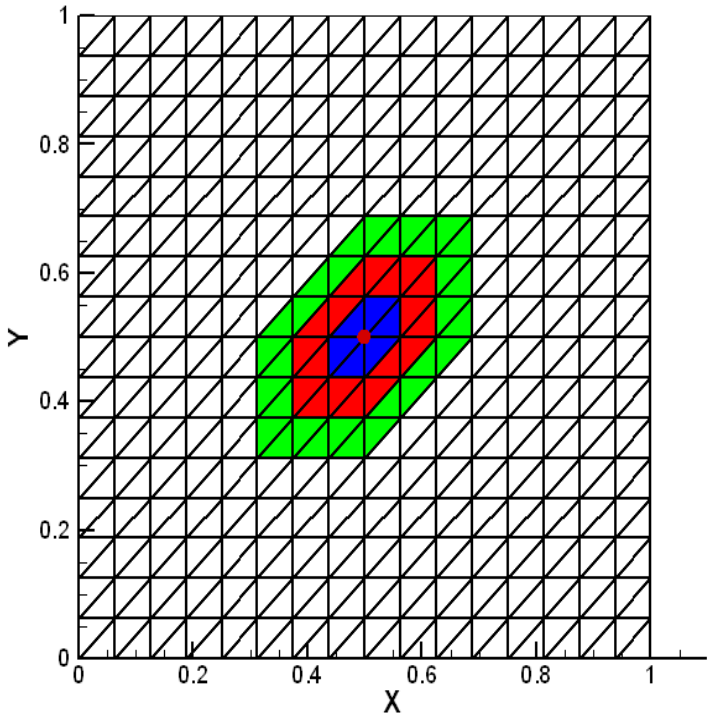

Figure 4.1: Local domain with oversampling. $\omega^{i,0}$=blue region, $\omega^{i,1}$=blue and red regions, $\omega^{i,2}$=blue, red and green regions.

where C_∞, C_G are two constants.

Based on above special partition of unity, we develop a new local and parallel variational mutiscale method as follows.

ALGORITHM PVMS-PU:

Step 1. Using the variational multiscale method to find a globally coarse grid solution $(\mathbf{u}_H, p_H) \in (\mathbf{X}_0^H, M_0^H)$ such that

$$\nu a(\mathbf{u}_H, \mathbf{v}) + b(\mathbf{u}_H, \mathbf{u}_H, \mathbf{v}) - d(\mathbf{v}, p_H) + d(\mathbf{u}_H, q) + G(\mathbf{u}_H, \mathbf{v}) = (\mathbf{f}, \mathbf{v}), \quad \forall (\mathbf{v}, q) \in (\mathbf{X}_0^H, M_0^H). \tag{4.5}$$

Step 2. For a given τ_{H_p}, fix $s \geqslant 1$, correct the residue $(\mathbf{e}^i, \epsilon^i)$ on a fine grid of each overlapping subdomain $\omega^{i,s}$ of τ_{H_p} in parallel, $(\mathbf{e}^i, \epsilon^i) \in (\mathbf{X}_0^h(\omega^{i,s}), M_0^h(\omega^{i,s}))$, $i = 1, 2, \cdots, N$, such that

$$\begin{aligned} \nu a(\mathbf{e^i}, \mathbf{v}) + \mathbf{b}(\mathbf{e^i}, \mathbf{u_H}, \mathbf{v}) + \mathbf{b}(\mathbf{u_H}, \mathbf{e^i}, \mathbf{v}) - \mathbf{d}(\mathbf{v}, \epsilon^{\mathbf{i}}) + \mathbf{d}(\mathbf{e^i}, \mathbf{q}) + \beta(\nabla \mathbf{e^i}, \nabla \mathbf{v}) \\ = (\mathcal{R}(\mathbf{u_H}, \mathbf{p_H}), \mathbf{v}), \quad \forall (\mathbf{v}, \mathbf{q}) \in (\mathbf{X_0^h}(\omega^{\mathbf{i,s}}), \mathbf{M_0^h}(\omega^{\mathbf{i,s}})), \end{aligned} \tag{4.6}$$

where $(\mathcal{R}(\mathbf{u_H}, \mathbf{p_H}), \mathbf{v}) = (\mathbf{f}, \mathbf{v}) - \nu(\mathbf{u_H}, \mathbf{v}) - \mathbf{b}(\mathbf{u_H}, \mathbf{u_H}, \mathbf{v}) + \mathbf{d}(\mathbf{v}, \mathbf{p_H}) - \mathbf{d}(\mathbf{u_H}, \mathbf{q})$. Here

$$\begin{aligned} \mathbf{X}_0^h(\omega^{i,s}) &:= \{\mathbf{v} \in \mathbf{X}^h(\Omega) : \text{ supp } \mathbf{v} \subset\subset \omega^{i,s}\}, \\ M_0^h(\omega^{i,s}) &:= \{q \in M^h(\Omega) : \text{ supp } q \subset\subset \omega^{i,s} \text{and } \int_{\omega^{i,s}} q \mathtt{d}x = 0\}. \end{aligned}$$

Step 3. Update: $(\mathbf{u}^i, p^i) = (\mathbf{u}_H, p_H) + (\mathbf{e}^i, \epsilon^i)$ in $\omega^{i,s}$.

Step 4. Obtain the finite element solution $\mathbf{u}^h = \sum_{i=1}^{N} \varphi_i \mathbf{u}^i$, $p^h = \sum_{i=1}^{N} \varphi_i p^i$.

In order to get the error estimate of this algorithm we first introduce a lemma and regularity property from [36] and we list as two lemmas here.

Lemma 4.1. *Suppose that* $\mathbf{g} \in \mathbf{H^{-1}}(\mathbf{\Omega})^{\mathbf{n}}, \mathbf{0} < \mathbf{H} \leqslant \mathbf{\bar{h}_0}$ *and* $S \subset\subset \Omega_0 \subset \Omega$. *Then* $(\mathbf{w}, \mathbf{r}) \in \mathbf{X_h}(\mathbf{\Omega}) \times \mathbf{M_h}(\mathbf{\Omega})$ *defined by*

$$\begin{aligned} \nu a(\mathbf{w}, \mathbf{v}) + (\mathbf{u_h}, \mathbf{w}, \mathbf{v}) + \mathbf{b}(\mathbf{w}, \mathbf{u_H}, \mathbf{v}) - \mathbf{d}(\mathbf{v}, \mathbf{r}) + \mathbf{d}(\mathbf{w}, \mathbf{q}) = (\mathbf{g}, \mathbf{v}), \\ \forall (\mathbf{v}, \mathbf{q}) \in \mathbf{X_0^h}(\mathbf{\Omega_0}) \times \mathbf{M_0^h}(\mathbf{\Omega_0}) \end{aligned} \tag{4.7}$$

satisfies

$$\|\mathbf{w}\|_{\mathbf{1,D}} + \|\mathbf{r}\|_{\mathbf{0,D}} \leqslant \mathbf{C}(\|\mathbf{w}\|_{\mathbf{0,\Omega_0}} + \|\mathbf{r}\|_{\mathbf{-1,\Omega_0}} + \|\mathbf{g}\|_{\mathbf{-1,\Omega_0}}). \tag{4.8}$$

Lemma 4.2. *There exists a unique* $(\Psi_h, \Psi_h) \in X_h^0(\Omega) \times M_h^0(\Omega)$ *satisfying the dual problem:*

$$\begin{aligned}(\nu+\beta)a(\mathbf{v}, \mathbf{\Phi_h}) + \mathbf{b}(\mathbf{u_h}, \mathbf{v}, \mathbf{\Phi_h}) + \mathbf{b}(\mathbf{v}, \mathbf{u_h}, \mathbf{\Phi_h}) + \mathbf{d}(\mathbf{v}, \mathbf{\Psi_h}) - \mathbf{d}(\mathbf{\Phi_h}, \mathbf{q}) \\ = (\psi, \mathbf{v}) + (\phi, \mathbf{q}), \quad \forall (\mathbf{v}, \mathbf{q}) \in \mathbf{H_0^1(\Omega)} \times \mathbf{L_0^2(\Omega)}\end{aligned} \tag{4.9}$$

and has the following estimates

$$\|\Phi - \Phi_h\|_{1,\Omega} + \|\Psi - \Psi_h\|_{0,\Omega} \leqslant Ch(\|\phi\|_{0,\Omega} + \|\psi\|_{1,\Omega}),$$

$$\|\Phi_h\|_{1,\Omega} + \|\Psi_h\|_{0,\Omega} \leqslant C(\|\phi\|_{0,\Omega} + \|\psi\|_{1,\Omega}).$$

We also need the following lemma which can be proved easily so that we don't show the details here.

Lemma 4.3. *Let* $C_0 > 0$ *be a constant, and* $\{\varphi_i\}_1^N$ *be the partition of unity based on* τ_{H_p} *with* $H_p \geqslant C_0$. *Then there exist constant* C_1, C_2, C_3, C_4 *independent of* N

satisfying the following inequalities which will be used in the proof of next theorem.

$$\|\sum_{i=1}^{N}\varphi_i\mathbf{v}\|_{\mathbf{1},\mathbf{\Omega}} \leqslant \mathbf{C_1}\sum_{\mathbf{i=1}}^{\mathbf{N}}\|\varphi_\mathbf{i}\mathbf{v}\|_{\mathbf{1},\mathbf{\Omega}}, \quad \forall \in \mathbf{H^1}(\mathbf{\Omega}), \tag{4.10}$$

$$\|\sum_{i=1}^{N}\varphi_i q\|_{0,\Omega} \leqslant C_2\sum_{i=1}^{N}\|\varphi_i q\|_{0,\Omega}, \quad \forall q \in L^2(\Omega), \tag{4.11}$$

$$\|\mathbf{v}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega}} + \|\mathbf{q}\|^{\mathbf{2}}_{\mathbf{0},\mathbf{\Omega}} \leqslant \mathbf{C_3}(\sum_{\mathbf{i=1}}^{\mathbf{N}}(\|\varphi_\mathbf{i}\mathbf{v}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega}} + \|\varphi_\mathbf{i}\mathbf{q}\|^{\mathbf{2}}_{\mathbf{0},\mathbf{\Omega}})), \tag{4.12}$$

$$\|\varphi_i\mathbf{v}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega}} + \|\varphi_\mathbf{i}\mathbf{q}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega}} \leqslant \mathbf{C_4}(\|\mathbf{v}\|^{\mathbf{2}}_{\mathbf{1},\omega^\mathbf{i}} + \|\mathbf{q}\|^{\mathbf{2}}_{\mathbf{1},\omega^\mathbf{i}}). \tag{4.13}$$

Then we can prove the following theorem.

Theorem 4.1. *Assume that the conditions of Theorem 2.2 hold,* $0 < h \leqslant H$*, for a given* $H_p \geqslant C_0$ *and* $s \geqslant 1$*, the solution* $(\mathbf{u^h}, \mathbf{p^h})$ *defined by Algorithm PVMS-PU satisfies*

$$\|\mathbf{u_h} - \mathbf{u^h}\|_{\mathbf{1},\mathbf{\Omega}} + \|\mathbf{p_h} - \mathbf{p^h}\|_{\mathbf{0},\mathbf{\Omega}} \leqslant \mathbf{C}(\mathbf{H^3} + \mathbf{H}\alpha_\mathbf{H} + \alpha^{\mathbf{2}}_\mathbf{H} + \beta(\mathbf{H^2} + \alpha_\mathbf{H})). \tag{4.14}$$

Proof. **Step1**. In order to get the final result, let $D = \omega^i$, $\Omega_0 = \Omega^{i,s}$, then, first estimate $\|\mathbf{e^i}\|_{\mathbf{0},\mathbf{\Omega_o}} + \|\epsilon^\mathbf{i}\|_{-\mathbf{1},\mathbf{\Omega_o}}$. From (2.4) and (4.23) we can easily get the equality:

$$\begin{aligned}
&\nu a(\mathbf{u_h} - \mathbf{u_H}, \mathbf{\Phi_H}) + \mathbf{b}(\mathbf{u_h} - \mathbf{u_H}, \mathbf{u_H}, \mathbf{\Phi_H}) + \mathbf{b}(\mathbf{u_H}, \mathbf{u_h} - \mathbf{u_H}, \mathbf{\Phi_H}) \\
&\quad + b(\mathbf{u_h} - \mathbf{u_H}, \mathbf{u_h} - \mathbf{u_H}, \mathbf{\Phi_H}) - \mathbf{d}(\mathbf{\Phi_H}, \mathbf{p_h} - \mathbf{p_H}) + \mathbf{d}(\mathbf{u_h} - \mathbf{u_H}, \mathbf{\Psi_H}) \\
&\quad + G(\mathbf{u_h}, , \mathbf{\Phi_H}) - \mathbf{G}(\mathbf{u_H}, \mathbf{\Phi_H}) = \mathbf{0}.
\end{aligned} \tag{4.15}$$

From (2.4) and (4.24) we can obtain that

$$
\begin{aligned}
&(\nu+\beta)a(\mathbf{e^i},\mathbf{v})+\mathbf{b}(\mathbf{e^i},\mathbf{u_H},\mathbf{v})+\mathbf{b}(\mathbf{u_H},\mathbf{e^i},\mathbf{v})-\mathbf{d}(\mathbf{v},\epsilon^{\mathbf{i}})+\mathbf{d}(\mathbf{e^i},\mathbf{q})\\
&=\nu a(\mathbf{u_h}-\mathbf{u_H},\mathbf{v})+\mathbf{b}(\mathbf{u_h}-\mathbf{u_H},\mathbf{u_H},\mathbf{v})+\mathbf{b}(\mathbf{u_H},\mathbf{u_h}-\mathbf{u_H},\mathbf{v})\\
&\quad+b(\mathbf{u_h}-\mathbf{u_H},\mathbf{u_h}-\mathbf{u_H},\mathbf{v})-\mathbf{d}(\mathbf{v},\mathbf{p_h}-\mathbf{p_H})+\mathbf{d}(\mathbf{u_h}-\mathbf{u_H},\mathbf{q})+\mathbf{G}(\mathbf{u_h},\mathbf{v}).
\end{aligned}
\tag{4.16}
$$

We can get the following dual problem:

$$
\begin{aligned}
&(\phi,\mathbf{e^i})+(\psi,\epsilon^{\mathbf{i}})\\
&=(\nu+\beta)a(\mathbf{e^i},\mathbf{\Phi_h})+\mathbf{b}(\mathbf{u_h},\mathbf{e^i},\mathbf{\Phi_h})+\mathbf{b}(\mathbf{e^i},\mathbf{u_H},\mathbf{\Phi_h})+\mathbf{d}(\mathbf{e^i},\mathbf{\Psi_h})-\mathbf{d}(\mathbf{\Phi_h},\epsilon^{\mathbf{i}})\\
&=\nu a(\mathbf{u_h}-\mathbf{u_H},\mathbf{\Phi_h})+\mathbf{b}(\mathbf{u_h}-\mathbf{u_H},\mathbf{u_H},\mathbf{\Phi_h})+\mathbf{b}(\mathbf{u_H},\mathbf{u_h}-\mathbf{u_H},\mathbf{\Phi_h})\\
&\quad+b(\mathbf{u_h}-\mathbf{u_H},\mathbf{u_h}-\mathbf{u_H},\mathbf{\Phi_h})-\mathbf{d}(\mathbf{\Phi_h},\mathbf{p_h}-\mathbf{p_H})+\mathbf{d}(\mathbf{u_h}-\mathbf{u_H},\mathbf{\Psi_h})+\mathbf{G}(\mathbf{u_h},\mathbf{\Phi_h})\\
&=\nu a(\mathbf{u_h}-\mathbf{u_H},\mathbf{\Phi_h}-\mathbf{\Phi_H})+\mathbf{b}(\mathbf{u_h}-\mathbf{u_H},\mathbf{u_H},\mathbf{\Phi_h}-\mathbf{\Phi_H})\\
&\quad+b(\mathbf{u_H},\mathbf{u_h}-\mathbf{u_H},\mathbf{\Phi_h}-\mathbf{\Phi_H})+\mathbf{b}(\mathbf{u_h}-\mathbf{u_H},\mathbf{u_h}-\mathbf{u_H},\mathbf{\Phi_h}-\mathbf{\Phi_H})\\
&\quad-d(\Phi_h-\Phi_H,p_h-p_H)+d(\mathbf{u_h}-\mathbf{u_H},\mathbf{\Psi_h}-\mathbf{\Psi_H})\\
&\quad+G(\mathbf{u_h},\mathbf{\Phi_h}-\mathbf{\Phi_H})+\mathbf{G}(\mathbf{u_H},\mathbf{\Phi_H}).
\end{aligned}
\tag{4.17}
$$

Bound every item by using the properties of $a(\mathbf{u},\mathbf{v})$, $b(\mathbf{u},\mathbf{v},\mathbf{w})$, $d(\mathbf{v},p)$ in 2.1 and

Lemma 4.2:

$$\nu a(\mathbf{u_h}-\mathbf{u_H},\mathbf{\Phi_h}-\mathbf{\Phi_H}) \leqslant \mathbf{CH}\|\mathbf{u_h}-\mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}}(\|\phi\|_{\mathbf{0},\mathbf{\Omega_0}}+\|\psi\|_{\mathbf{1},\mathbf{\Omega_0}}),$$

$$b(\mathbf{u_h}-\mathbf{u_H},\mathbf{u_H},\mathbf{\Phi_h}-\mathbf{\Phi_H})+\mathbf{b}(\mathbf{u_H},\mathbf{u_h}-\mathbf{u_H},\mathbf{\Phi_h}-\mathbf{\Phi_H})$$

$$\leqslant CH\|\mathbf{u_h}-\mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}}(\|\phi\|_{\mathbf{0},\mathbf{\Omega_0}}+\|\psi\|_{\mathbf{1},\mathbf{\Omega_0}}),$$

$$b(\mathbf{u_h}-\mathbf{u_H},\mathbf{u_h}-\mathbf{u_H},\mathbf{\Phi_h}-\mathbf{\Phi_H}) \leqslant \mathbf{CH}\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega_0}}(\|\phi\|_{\mathbf{0},\mathbf{\Omega_0}}+\|\psi\|_{\mathbf{1},\mathbf{\Omega_0}}),$$

$$d(\Phi_h-\Phi_H,p_h-p_H) \leqslant CH\|p_h-p_H\|_{0,\Omega_0}(\|\phi\|_{0,\Omega_0}+\|\psi\|_{1,\Omega_0}),$$

$$d(\mathbf{u_h}-\mathbf{u_H},\mathbf{\Psi_h}-\mathbf{\Psi_H}) \leqslant \mathbf{CH}\|\mathbf{u_h}-\mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}}(\|\phi\|_{\mathbf{0},\mathbf{\Omega_0}}+\|\psi\|_{\mathbf{1},\mathbf{\Omega_0}}),$$

$$G(\mathbf{u_h},\mathbf{\Phi_h}-\mathbf{\Phi_H}) \leqslant \mathbf{C}\alpha_{\mathbf{h}}(\|\mathbf{u}-\mathbf{u_h}\|_{\mathbf{1},\mathbf{\Omega_0}}+\mathbf{h}\|\mathbf{u}\|_{\mathbf{2},\mathbf{\Omega_0}})(\|\phi\|_{\mathbf{0},\mathbf{\Omega_0}}+\|\psi\|_{\mathbf{1},\mathbf{\Omega_0}}),$$

$$G(\mathbf{u_H},\mathbf{\Phi_H}) \leqslant \mathbf{C}\alpha_{\mathbf{H}}(\|\mathbf{u}-\mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}}+\mathbf{H}\|\mathbf{u}\|_{\mathbf{2},\mathbf{\Omega_0}})(\|\phi\|_{\mathbf{0},\mathbf{\Omega_0}}+\|\psi\|_{\mathbf{1},\mathbf{\Omega_0}}).$$

Thanks to the above inequalities we can get

$$\begin{aligned}&(\phi,\mathbf{e^i})+(\psi,\epsilon^{\mathbf{i}})\\ &\leqslant C(H\|\mathbf{u_h}-\mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}}+\mathbf{H}\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega_0}}+\mathbf{H}\|\mathbf{p_h}-\mathbf{p_H}\|_{\mathbf{0},\mathbf{\Omega_0}}+\alpha_{\mathbf{h}}\|\mathbf{u}-\mathbf{u_h}\|_{\mathbf{1},\mathbf{\Omega_0}}\\ &\quad+\alpha_H\|\mathbf{u}-\mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}}+(\alpha_{\mathbf{h}}\mathbf{h}+\alpha_{\mathbf{H}}\mathbf{H})\|\mathbf{u}\|_{\mathbf{2},\mathbf{\Omega_0}})(\|\phi\|_{\mathbf{0},\mathbf{\Omega_0}}+\|\psi\|_{\mathbf{1},\mathbf{\Omega_0}}).\end{aligned} \tag{4.18}$$

Thus the following conclusion can be obtained:

$$\begin{aligned}&\|\mathbf{e^i}\|_{\mathbf{0},\mathbf{\Omega_0}} + \|\epsilon^{\mathbf{i}}\|_{-\mathbf{1},\mathbf{\Omega_0}}\\ &\leqslant C(H\|\mathbf{u_h} - \mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}} + \mathbf{H}\|\mathbf{p_h} - \mathbf{p_H}\|_{\mathbf{0},\mathbf{\Omega_0}} + \mathbf{H^3}\|\mathbf{u}\|_{\mathbf{2},\mathbf{\Omega_0}}).\end{aligned} \tag{4.19}$$

Step2. From (2.4) and the Algorithm we can get the following equality:

$$\begin{aligned}&(\nu+\beta)a(\mathbf{u_h} - \mathbf{u^h}, \mathbf{v}) + \mathbf{b}(\mathbf{u_h} - \mathbf{u^h}, \mathbf{u_H}, \mathbf{v}) + \mathbf{b}(\mathbf{u_H}, \mathbf{u_h} - \mathbf{u^h}, \mathbf{v}) - \mathbf{d}(\mathbf{v}, \mathbf{p_h} - \mathbf{p^h})\\ &+ d(\mathbf{u_h} - \mathbf{u^h}, \mathbf{q}) = -\mathbf{G}(\mathbf{u_h}, \mathbf{v}) + \beta\mathbf{a}(\mathbf{u_h} - \mathbf{u_H}, \mathbf{v}) - \mathbf{b}(\mathbf{u_h} - \mathbf{u_H}, \mathbf{u_h} - \mathbf{u_H}, \mathbf{v}).\end{aligned}$$

Then using Lemma 4.1 and (4.19) we can obtain:

$$\begin{aligned}&\|\mathbf{u_h} - \mathbf{u^i}\|_{\mathbf{1},\mathbf{D}} + \|\mathbf{p_h} - \mathbf{p^i}\|_{\mathbf{0},\mathbf{D}}\\ &\leqslant C(\|\mathbf{u_h} - \mathbf{u^i}\|_{\mathbf{0},\mathbf{\Omega_0}} + \|\mathbf{p_h} - \mathbf{p^i}\|_{-\mathbf{1},\mathbf{\Omega_0}} + \|\mathbf{u_h} - \mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega_0}} + \beta\|\mathbf{u_h} - \mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}}\\ &\quad + \alpha_h\|\mathbf{u_h}\|_{\mathbf{1},\mathbf{\Omega_0}})\\ &\leqslant C(\|\mathbf{u_h} - \mathbf{u_H}\|_{\mathbf{0},\mathbf{\Omega_0}} + \|\mathbf{p_h} - \mathbf{p_H}\|_{-\mathbf{1},\mathbf{\Omega_0}} + \|\mathbf{e^i}\|_{\mathbf{0},\mathbf{\Omega_0}} + \|\epsilon^{\mathbf{i}}\|_{-\mathbf{1},\mathbf{\Omega_0}} + \|\mathbf{u_h} - \mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega_0}}\\ &\quad + \beta\|\mathbf{u_h} - \mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}} + \alpha_{\mathbf{h}}\|\mathbf{u_h}\|_{\mathbf{1},\mathbf{\Omega_0}})\\ &\leqslant C(\|\mathbf{u_h} - \mathbf{u_H}\|_{\mathbf{0},\mathbf{\Omega_0}} + \|\mathbf{p_h} - \mathbf{p_H}\|_{-\mathbf{1},\mathbf{\Omega_0}} + \|\mathbf{u_h} - \mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega_0}} + \mathbf{H}\|\mathbf{u_h} - \mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}}\\ &\quad + H\|p_h - p_H\|_{0,\Omega_0} + H^3\|\mathbf{u}\|_{\mathbf{2},\mathbf{\Omega_0}} + \beta\|\mathbf{u_h} - \mathbf{u_H}\|_{\mathbf{1},\mathbf{\Omega_0}} + \alpha_{\mathbf{h}}\|\mathbf{u_h}\|_{\mathbf{1},\mathbf{\Omega_0}}).\end{aligned} \tag{4.20}$$

Thus, the inequality (4.20) stands for every $D = \omega^i, \Omega_0 = \omega^{i,s},\ i = 1, 2, \cdots, N$.

We then get the global error estimate by using Theorem 1 and Lemma 2:

$$
\begin{aligned}
&\|\mathbf{u_h}-\mathbf{u^h}\|_{\mathbf{1},\mathbf{\Omega}}+\|\mathbf{p_h}-\mathbf{p^h}\|_{\mathbf{0},\mathbf{\Omega}}\\
&=\|\sum_{i=1}^{N}\varphi_i(\mathbf{u_h}-\mathbf{u^i})\|_{\mathbf{1},\mathbf{\Omega}}+\|\sum_{\mathbf{i=1}}^{\mathbf{N}}\varphi_{\mathbf{i}}(\mathbf{p_h}-\mathbf{p^i})\|_{\mathbf{0},\mathbf{\Omega}}\\
&\leqslant C(\sum_{i=1}^{N}\|\varphi_i(\mathbf{u_h}-\mathbf{u^i})\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega}}+\sum_{\mathbf{i=1}}^{\mathbf{N}}\|\varphi_{\mathbf{i}}(\mathbf{p_h}-\mathbf{p^i})\|^{\mathbf{2}}_{\mathbf{0},\mathbf{\Omega}})^{\frac{\mathbf{1}}{\mathbf{2}}}\\
&\leqslant C(\sum_{i=1}^{N}\|\varphi_i(\mathbf{u_h}-\mathbf{u^i})\|^{\mathbf{2}}_{\mathbf{1},\omega^{\mathbf{i}}}+\sum_{\mathbf{i=1}}^{\mathbf{N}}\|\varphi_{\mathbf{i}}(\mathbf{p_h}-\mathbf{p^i})\|^{\mathbf{2}}_{\mathbf{0},\omega^{\mathbf{i}}})^{\frac{\mathbf{1}}{\mathbf{2}}}\\
&\leqslant C(\sum_{i=1}^{N}\|\mathbf{u_h}-\mathbf{u^i}\|^{\mathbf{2}}_{\mathbf{1},\omega^{\mathbf{i}}}+\sum_{\mathbf{i=1}}^{\mathbf{N}}\|\mathbf{p_h}-\mathbf{p^i}\|^{\mathbf{2}}_{\mathbf{0},\omega^{\mathbf{i}}})^{\frac{\mathbf{1}}{\mathbf{2}}}\\
&\leqslant C(\sum_{i=1}^{N}(\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{0},\omega^{\mathbf{i,s}}}+\|\mathbf{p_h}-\mathbf{p_H}\|^{\mathbf{2}}_{-\mathbf{1},\omega^{\mathbf{i,s}}}+\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{4}}_{\mathbf{1},\omega^{\mathbf{i,s}}}\\
&\quad+H^2\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\omega^{\mathbf{i,s}}}+\mathbf{H^2}\|\mathbf{p_h}-\mathbf{p_H}\|^{\mathbf{2}}_{\mathbf{0},\omega^{\mathbf{i,s}}}+\mathbf{H^6}\|\mathbf{u}\|^{\mathbf{2}}_{\mathbf{2},\omega^{\mathbf{i,s}}}\\
&\quad+\beta^2\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\omega^{\mathbf{i,s}}}+\alpha^{\mathbf{2}}_{\mathbf{h}}\|\mathbf{u_h}\|^{\mathbf{2}}_{\mathbf{1},\omega^{\mathbf{i,s}}}))^{\frac{\mathbf{1}}{\mathbf{2}}}\\
&\leqslant C(\sum_{i=1}^{N}\sum_{E_i\in\omega^{i,s}}(\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{0},\mathbf{E_i}}+\|\mathbf{p_h}-\mathbf{p_H}\|^{\mathbf{2}}_{-\mathbf{1},\mathbf{E_i}}+\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{4}}_{\mathbf{1},\mathbf{E_i}}\\
&\quad+H^2\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{E_i}}+\mathbf{H^2}\|\mathbf{p_h}-\mathbf{p_H}\|^{\mathbf{2}}_{\mathbf{0},\mathbf{E_i}}+\mathbf{H^6}\|\mathbf{u}\|^{\mathbf{2}}_{\mathbf{2},\mathbf{E_i}}\\
&\quad+\beta^2\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{E_i}}+\alpha^{\mathbf{2}}_{\mathbf{h}}\|\mathbf{u_h}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{E_i}}))^{\frac{\mathbf{1}}{\mathbf{2}}}\\
&\leqslant CC_{ov}(\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{0},\mathbf{\Omega}}+\|\mathbf{p_h}-\mathbf{p_H}\|^{\mathbf{2}}_{-\mathbf{1},\mathbf{\Omega}}+\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{4}}_{\mathbf{1},\mathbf{\Omega}}+\mathbf{H^2}\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega}}\\
&\quad+H^2\|p_h-p_H\|^2_{0,\Omega}+H^6\|\mathbf{u}\|^{\mathbf{2}}_{\mathbf{2},\mathbf{\Omega}}+\beta^{\mathbf{2}}\|\mathbf{u_h}-\mathbf{u_H}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega}}+\alpha^{\mathbf{2}}_{\mathbf{h}}\|\mathbf{u_h}\|^{\mathbf{2}}_{\mathbf{1},\mathbf{\Omega}})^{\frac{\mathbf{1}}{\mathbf{2}}}\\
&\leqslant C(H^3+H\alpha_H+\alpha_H^2+\beta(H^2+\alpha_H)).
\end{aligned}
\tag{4.21}
$$

Here, C_{ov} is a finite integer defined as the maximal number of elements E_j contained in each subdomain $\omega^{i,s}$. It is determined by the layer index s, the minimum angle of the regular triangulation τ_{H_p}, and is independent of N. □

Remark 4.1. *It has been mentioned in [89] that α should been chosen as $O(h^2)$ in the computation. Thus we can make a conclusion that β only needs to be $O(H)$ to keep the rate of convergence.*

Using the triangle inequality we can get the following theorem directly from theorem 1 and theorem 2.

Theorem 4.2. *Assume that the conditions of Theorem 2.2 hold, $0 < h \leqslant H$, for a given $H_p \geqslant C_0$ and $s \geqslant 1$, choose $\alpha_H = O(H^2)$, $\beta = O(H)$, then, the solution $(\mathbf{u^h}, \mathbf{p^h})$ defined by Algorithm PVMS-PU satisfies*

$$\|\mathbf{u} - \mathbf{u^h}\|_{\mathbf{1},\mathbf{\Omega}} + \|\mathbf{p} - \mathbf{p^h}\|_{\mathbf{0},\mathbf{\Omega}} \leqslant \mathbf{C}(\mathbf{h^2} + \mathbf{H^3}). \tag{4.22}$$

4.1.2 Numerical tests

The algorithm in all experiments is implemented by the public finite element software Freefem++ [29]. All simulations were performed on a dawning parallel cluster composed of 32 nodes, each with eight-core 2.0 GHz CPU, 2 GB × 8 DRAM, and connected together by 20Gbps InfiniBand. The message-passing is supported by

MPICH.

Implementation

To verify the analysis results, we consider 2D numerical examples. Dividing Ω into sub-squares with equal sizes h (or H, H_p), and drawing the diagonal in each sub-square, we obtain the regular triangulation τ_h (or τ_H, τ_{H_p}).

For convenience of presentation, we introduce the following notations:

SFEM means the standard finite element method. Namely, the nonlinear systems are solved by Newton iteration.

GVMS means the finite element variational multiscale method based on two local Gauss integrations (2.4).

PVMS-PU means `ALGORITHM PVMS-PU`.

Rates of convergence study

Let $\Omega = [0,1]\times[0,1]$ and the exact solution of the stationary Navier-Stokes equations (2.1) be given by $(\mathbf{u} = (u_1, u_2), p)$:

$$u_1 = 10x^2(x-1)^2y(y-1)(2y-1),$$

$$u_2 = -10x(x-1)(2x-1)y^2(y-1)^2,$$

$$p = 10(2x-1)(2y-1),$$

here, $\nu = 1.0$ for simplicity, $\mathbf{f}$ and the boundary conditions are set by $(\mathbf{u} = (u_1, u_2), p)$.

To get the optimal orders for H^1-norm of velocity and L^2-norm of pressure, we should choose H and h such that $h \sim H^{\frac{3}{2}}$. In this example, we compute the finite element solutions by PVMS-PU with coarse mesh sizes $H = \frac{1}{12n}$ (n=1, 2, 3, 4) and the corresponding fine mesh sizes $h = H/m$ (m=4, 5, 6, 7). Besides, according to Theorem 2.2and 4.2, we choose $\alpha_H = 0.1H^2$, $\beta = 0.1H$. The corresponding linear algebraic system is solved by LU factorization. Convergence of the Newton iteration is achieved when the relative H^1-error of successive iterative velocities is within a fixed tolerance of 10^{-6}, i.e., the following condition is satisfied:

$$\frac{||\mathbf{u}_\mu^{n+1} - \mathbf{u}_\mu^n||_{1,\Omega}}{||\mathbf{u}_\mu^{n+1}||_{1,\Omega}} \leqslant 10^{-6},$$

where $\mathbf{u}_\mu^n$ (μ could be h, H) is the nth Newton iterative solution.

For PVMS-PU, we fix P_1-PU on τ_{H_p}, $H_p = 1/12$, $s = 1$, thus, $N = 169$, all simulations are implemented with 32 processors.

Table 4.1: The errors of GVMS

h	$\|\|\mathbf{u} - \bar{\mathbf{u}}_h\|\|_{1,\Omega}$	Order	$\|\|p - \bar{p}_h\|\|_{0,\Omega}$	Order	CPU
1/48	0.000365636	-	0.00112071	-	3.62
1/120	5.89085e-05	1.9943	0.00017977	1.99814	33.46
1/216	1.92045e-05	1.99225	5.68389e-05	1.98735	153.99
1/336	-	-	-	-	-

To further test our PVMS-PU, we also consider another problem (referred as

Table 4.2: The errors of PVMS-PU

H	h	$\|\|\mathbf{u}-\mathbf{u}^h\|\|_{1,\Omega}$	Order	$\|\|p-p^h\|\|_{0,\Omega}$	Order	Wall time
1/12	1/48	0.00102771	-	0.00118434	-	2.85
1/24	1/120	0.000126349	2.05336	0.00018409	1.99975	12.17
1/36	1/216	3.57543e-05	2.11658	5.64822e-05	2.01571	50.62
1/48	1/336	1.47535e-05	2.05336	2.35785e-05	1.99975	182.75

Solution 2) with exact solution $\mathbf{u} = (u_1, u_2)$

$$u_1 = \sin(\pi x)^2 \sin(2\pi y),$$

$$u_2 == -\sin(2\pi x)\sin(\pi y)^2,$$

$$p = \cos(\pi x)\cos(\pi y).$$

In present computations, the same parameters h, H and s for PVMS-PU are chosen as Solution 1. The results are tabulated in Tables 4.3 and 4.4. From these two tables, we can observe similar phenomena and draw same conclusion as found from Tables 4.10 and 4.2.

Table 4.3: Solution 2, the errors of GVMS

h	$\|\|\mathbf{u}-\bar{\mathbf{u}}_h\|\|_{1,\Omega}$	Order	$\|\|p-\bar{p}_h\|\|_{0,\Omega}$	Order	Wall time
1/48	0.00566745	-	0.000179236	-	3.32
1/120	0.00090770	1.99891	2.86259e-05	2.00198	29.82
1/216	0.000280193	1.99977	9.00352e-06	1.96788	153.65
1/336	-	-	-	-	-

Table 4.4: Solution 2, the errors of PVMS-PU

H	h	$\|\|\mathbf{u}-\mathbf{u}^h\|\|_{1,\Omega}$	Order	$\|\|p-p^h\|\|_{0,\Omega}$	Order	Wall time
1/12	1/48	0.00809542	-	0.00283574	-	5.82
1/24	1/120	0.000937267	2.35306	0.00019754	2.9075	13.33
1/36	1/216	0.000284110	2.03068	4.44115e-05	2.53909	51.48
1/48	1/336	0.000116765	2.01252	1.56627e-05	2.35885	193.94

The driven cavity flow

A popular benchmark problem for testing numerical schemes is the 'fluid driven cavity'. This problem is chosen because some benchmark data is available for comparison. In this problem, computations are carried out in the domain $\Omega = [0,1] \times [0,1]$. Flow is driven by the tangential velocity field applied to the top boundary in the absence of other body forces. On the top side $\{(x,1) : 0 < x < 1\}$, the velocity is equal to $\mathbf{u} = (1,0)$, and zero Dirichlet conditions are imposed on the rest of the boundary.

Based on PVMS-PU with P_1-PU on τ_{H_p}, $H_p = 1/12$, $s = 1$, we compute for Reynolds numbers $Re = 5000$ with fixed $H = 1/48$, $h = 96$ and $Re = 10000$ with fixed $H = 1/60$, $h = 120$, and $\alpha_H = 0.1H$, $\beta = 0.1H$. the computational results are shown in Fig 4.2, 4.3 and 4.4, 4.5, comparing with the results of Ghia, Ghia, and Shin [31]. Ghia et al.'s algorithm is based on the time dependent stream function using the coupled implicit and multigrid methods.

For different Reynolds numbers, the x component of velocity along the vertical centerline and y component of velocity along the horizontal centerlines by PUPVMS

are drawn in Fig 4.2, 4.3 and 4.4, 4.5. The accuracy of the computed solutions by PUPVMS has good agreements with the benchmark data of Ghia et al. [31].

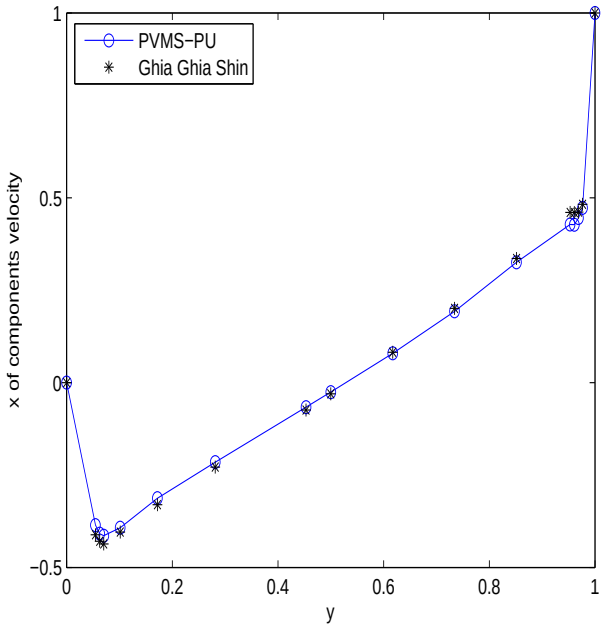

Figure 4.2: PUPVMS for $Re = 5000$, x component of velocity along the vertical centerline

.

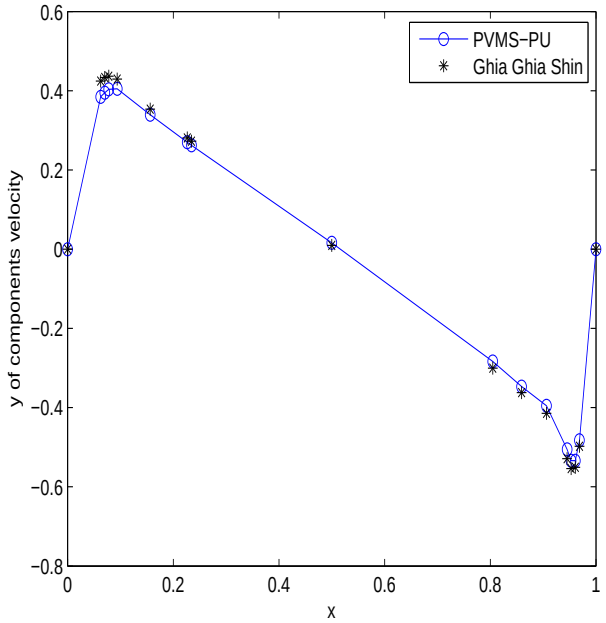

Figure 4.3: PUPVMS for $Re = 5000$, y component of velocity along the horizontal centerline.

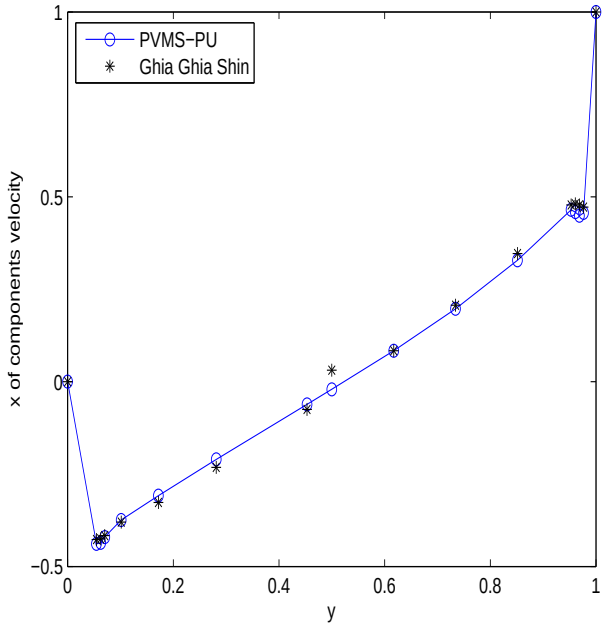

Figure 4.4: PUPVMS for $Re = 10000$, x component of velocity along the vertical centerline.

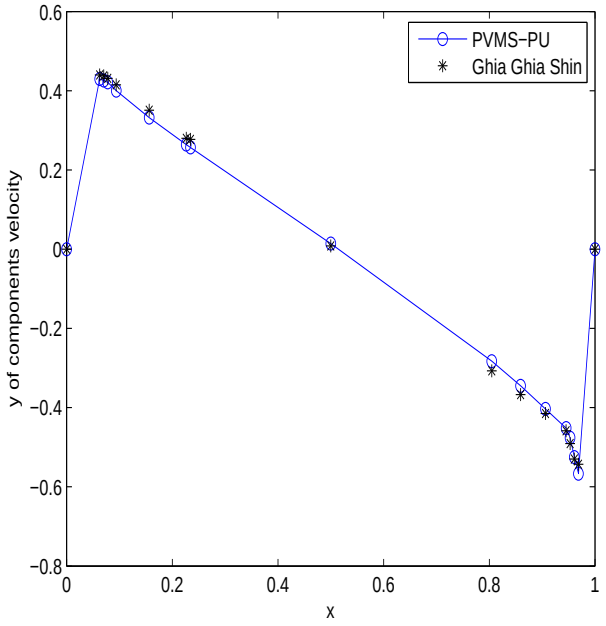

Figure 4.5: PUPVMS for $Re = 10000$, y component of velocity along the horizontal centerline.

4.2 Partition of unity parallel variational multiscale method

4.2.1 Algorithm

We develop a new local and parallel variational multiscale method as follows.

ALGORITHM PUPVMS:

Step 1. Using the variational multiscale method to find a global coarse grid solution $(\mathbf{u}_H, p_H) \in (\mathbf{X}_0^H, M_0^H)$ such that

$$\mathcal{C}(\mathbf{u}_H, p_H; \mathbf{v}, q) + b(\mathbf{u}_H, \mathbf{u}_H, \mathbf{v}) + G(\mathbf{u}_H, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall (\mathbf{v}, q) \in (\mathbf{X}_0^H, M_0^H). \quad (4.23)$$

Step 2. Correct the residue $(\mathbf{e}^i, \epsilon^i)$ on a fine grid of each overlapping subdomain $\omega^{i,s}$ in parallel, $(\mathbf{e}^i, \epsilon^i) \in (\mathbf{X}_0^h(\omega^{i,s}), M_0^h(\omega^{i,s}))$, $i = 1, 2, \cdots, N$, such that

$$\begin{aligned} &\mathcal{C}(\mathbf{e}^i, \epsilon^i; \mathbf{v}, q) + b(\mathbf{e}^i, \mathbf{u}_H, \mathbf{v}) + b(\mathbf{u}_H, \mathbf{e}^i, \mathbf{v}) + G(\mathbf{e}^i, \mathbf{v}) \\ &= (\mathbf{f}, \mathbf{v}) - \mathcal{C}(\mathbf{u}_H, p_H; \mathbf{v}, q) - b(\mathbf{u}_H, \mathbf{u}_H, \mathbf{v}) \quad \forall (\mathbf{v}, q) \in (\mathbf{X}_0^h(\omega^{i,s}), M_0^h(\omega^{i,s})). \end{aligned} \tag{4.24}$$

Step 4. Obtain the globally residue $\mathbf{e}^h = \sum_{i=1}^{N} \phi_i \mathbf{e}^i$, $\epsilon^h = \sum_{i=1}^{N} \phi_i \epsilon^i$.

Step 5. Update and yield the globally finite element solution $(\mathbf{u}^h, p^h) = (\mathbf{u}_H, p_H) + (\mathbf{e}^h, \epsilon^h)$.

Remark 1: Based on the above algorithm we have two ways to refine the local domains.

Refinement 1

For any i, s, straightly uniform refine the sub-domain $\omega^{i,s}$ with meshsize h, see the up figure of Fig 4.6, 4.7.

Refinement 2

Noting that, in Step 4, $\mathbf{e}^h = \sum_{i=1}^{N} \phi_i \mathbf{e}^i$, $\epsilon^h = \sum_{i=1}^{N} \phi_i \epsilon^i$, and $\mathtt{supp}\phi_i \subset \overline{\omega^{i,0}}$, which means that we only use $\mathbf{e}^i$, ϵ^i in $\omega^{i,0}$ finally, therefore, different from *Refinement 1*, for any i, s, we only refine $\omega^{i,0}$ with meshsize h uniformly, and using the coarser mesh for the oversampling parts, the further away from $\omega^{i,0}$, the coarser the mesh.

Of course, here, we need refine somewhere close to $\omega^{i,0}$ to avoid the hang nodes, see the down figure of Fig 4.6, 4.7.

Remark 2: Here

$$\begin{aligned}
\mathbf{X}_0^h(\omega^{i,s}) &:= \{\mathbf{v} \in \mathbf{X}^h(\Omega) : \ \mathbf{v}|_{\Omega\backslash\omega^{i,s}=0}\}, \\
M_0^h(\omega^{i,s}) &:= \{q \in M^h(\Omega) : \int_{\omega^{i,s}} q\mathtt{d}x = 0 \ \mathtt{and} \quad q = 0 \ \mathtt{on} \ \partial\omega^{i,s}\backslash\partial\Omega \ \mathtt{and} \ \mathtt{outside} \ \omega^{i,s}\}.
\end{aligned}$$

Note that, we enforce zero Dirichlet boundary condition for both pressure and velocity. Such restriction won't lead to singular problems since the zero mean-value constraint for the pressure enforces a unique solution. This treatment will yields a better accuracy than that with the traditional local pressure space

$$M_0^h(\omega^{i,s}) := \{q \in M^h(\Omega) : \int_{\omega^{i,s}} q\mathtt{d}x = 0 \ \mathtt{and} \quad p \ \mathtt{has} \ \mathtt{support} \ \mathtt{in} \ \omega^{i,s}\}.$$

The similar boundary conditions are used and discussed in [71, 49, 62].

Remark 3: Note that, the sub-problems only depend on the global coarse grid solution, and are independent of each other. Thus they can be carried out in parallel perfectly.

Remark 4: In order to diminish the undesirable effect by the artificial homogeneous boundary conditions for the local subproblems, we usually ask for some oversampling, namely, require the index $s \geqslant 1$ of $\omega^{i,s}$.

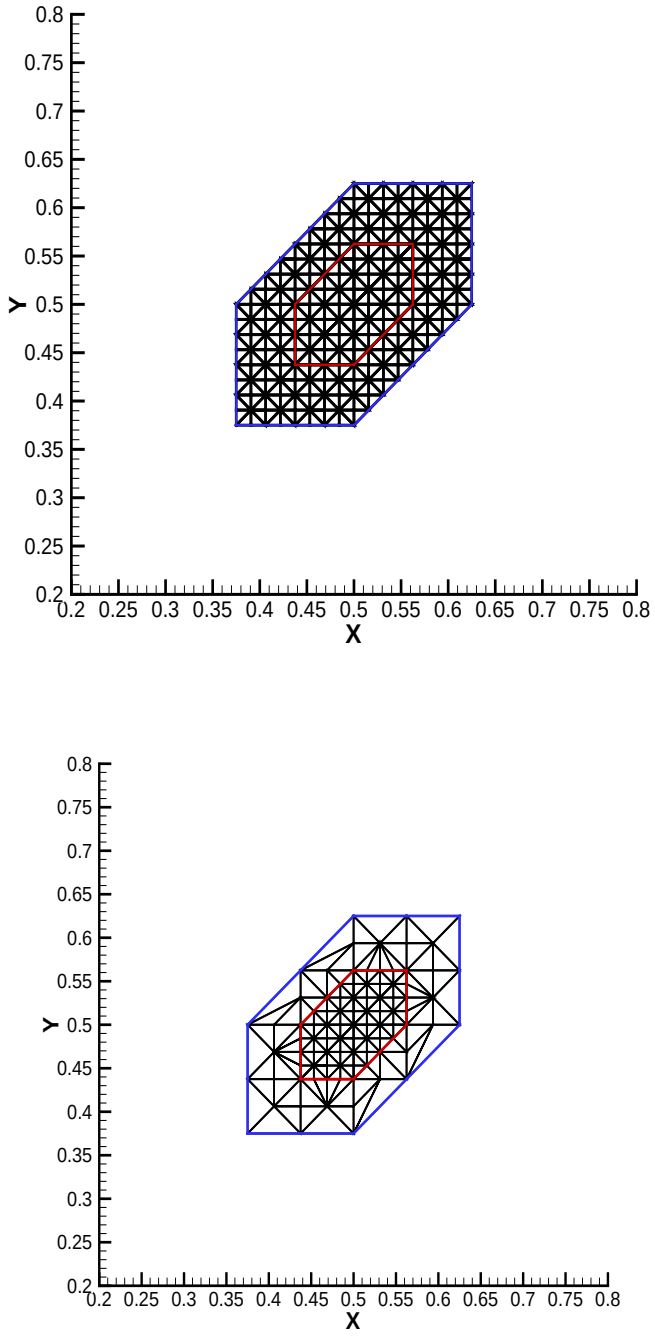

Figure 4.6: Up: Refinement 1 for $\omega^{i,1}$; down, Refinement 2 for $\omega^{i,1}$.

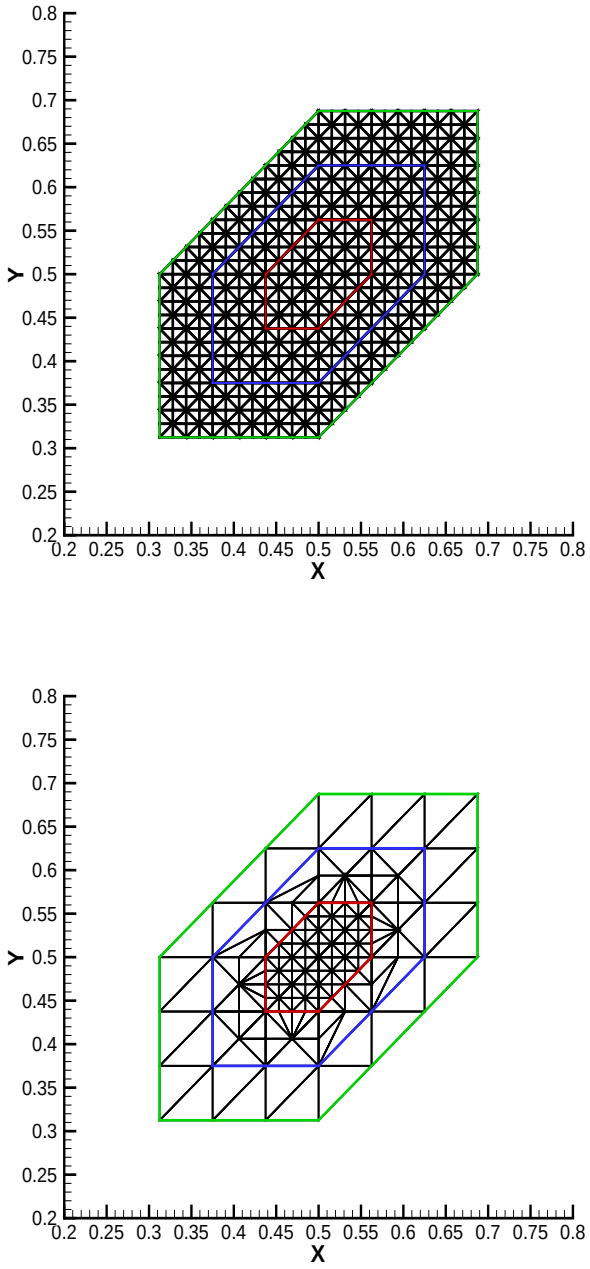

Figure 4.7: Up: Refinement 1 for $\omega^{i,2}$; down, Refinement 2 for $\omega^{i,2}$.

4.2.2 Numerical tests

The algorithm in all experiments is implemented by the public finite element software Freefem++ [29]. All simulations were performed on a dawning parallel cluster composed of 32 nodes, each with eight-core 2.0 GHz CPU, 2 GB × 8 DRAM, and connected together by 20Gbps InfiniBand. The message-passing is supported by MPICH.

Implementation

To verify the analysis results, we consider 2D numerical examples. Dividing Ω into sub-squares with equal sizes h (or H), and drawing the diagonal in each sub-square, we obtain the regular triangulation τ_h (or τ_H). The stable Taylor-Hood finite elements $(\mathbf{P}_2 - P_1)$ are used to solve the Navier-Stokes equations.

Thanks to [89], by $\mathbf{P}_2 - P_1$, we will use the equivalent formulation of (2.4), the finite element variational multiscale method based on two local Gauss integrations (which is more efficient) as follows: find $(\bar{\mathbf{u}}_h, \bar{p}_h) \in (\mathbf{X}_0^h, M_0^h)$ such that

$$\mathcal{C}(\bar{\mathbf{u}}_h, \bar{p}_h; \mathbf{v}, q) + b(\bar{\mathbf{u}}_h, \bar{\mathbf{u}}_h, \mathbf{v}) + G(\bar{\mathbf{u}}_h, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall (\mathbf{v}, q) \in (\mathbf{X}_0^h, M_0^h). \tag{4.25}$$

where,

$$G(\bar{\mathbf{u}}_h, \mathbf{v}) = \alpha \sum_{\Omega_e \in \tau_h} \{ \int_{\Omega_e, m} \nabla \bar{\mathbf{u}}_h \nabla \mathbf{v} \mathbf{d}x - \int_{\Omega_e, 1} \nabla \bar{\mathbf{u}}_h \nabla \mathbf{v} \mathbf{d}x \} \quad \forall \bar{\mathbf{u}}_h, \ \mathbf{v} \in \mathbf{X}^h.$$

Here $\int_{\Omega_e, i} g(x)\mathbf{d}x$ denotes an appropriate Gauss integral over Ω_e which is exact for polynomials of degree i, $i = 1, m(m > 1)$. For all test functions $\mathbf{v} \in \mathbf{X}^h$, $\nabla \bar{\mathbf{u}}_h$ must be piecewise constant when $i = 1$.

The corresponding linear algebraic system is solved by LU factorization. Convergence of the Newton iteration is achieved when the relative H^1-error of successive iterative velocities is within a fixed tolerance of 10^{-6}, i.e., the following condition is satisfied:

$$\frac{||\mathbf{u}_\mu^{n+1} - \mathbf{u}_\mu^n||_{1,\Omega}}{||\mathbf{u}_\mu^{n+1}||_{1,\Omega}} \leqslant 10^{-6},$$

where $\mathbf{u}_\mu^n$ (μ could be h, H) is the nth Newton iterative solution.

For convenience of presentation, we introduce the following notations:

SFEM means the standard finite element method. Namely, the nonlinear systems are solved by Newton iteration.

GVMS means the finite element variational multiscale method based on two local Gauss integrations (4.25).

Two-GVMS means the two-grid finite element variational multiscale method

based on two local Gauss integrations [52].

PUPVMS means `ALGORITHM PUPVMS`.

Local time means the CPU time of only solving sub-problems.

SwapData time means the CPU time for data communication in Step 4.

Wall time means the CPU time of solving coarse globally problem in Step 1 plus Local time and Swapdata time.

Problem 1

The first test problem is a problem in $\Omega = [0,1] \times [0,1]$, where the exact solution of the stationary Navier-Stokes equations (2.1) is given by ($\mathbf{u} = (u_1, u_2), p$):

$$u_1 = 10x^2(x-1)^2y(y-1)(2y-1),$$

$$u_2 = -10x(x-1)(2x-1)y^2(y-1)^2,$$

$$p = 10(2x-1)(2y-1).$$

Here, $\nu = 1.0$. For simplicity, $\mathbf{f}$ and the boundary conditions are set by ($\mathbf{u} = (u_1, u_2), p$). Note that, for $\nu = 1.0$, this ia a laminar case, which does not require stabilized method. We just use this very simple example to test some properties of PUPVMS.

Firstly, we examine the effect of the oversampling for PUPVMS. Table 4.5, 4.6

show the errors of PUPVMS with fixed H, h and processors, with different layers of oversampling at *Refinement 1* and *Refinement 2* respectively, and $\alpha = 0.1\mu^2$, $\mu = H$ in Step 1, $\mu = h$ in Step 2. From this two tables, we know that, PUPVMS without oversampling does not yield the acceptable accuracies usually by the effect of the artificial boundary conditions, when oversampling is introduced, $s \geqslant 1$, the accuracies get better. Compare PUPVMS at *Refinement 1* with PUPVMS at *Refinement 2*, they obtain the similar accuracies with same s, when $s \geqslant 2$, the improvement becomes slowly. However, the latter one cost less local time than the former. The local time for PUPVMS at *Refinement 1* increases quickly with the oversampling parameter s, however, the time for PUPVMS at *Refinement 2* increases slowly, since PUPVMS at *Refinement 2* uses coarse mesh outside of $\omega^{i,0}$, and introduces less variables than that of *Refinement 1*.

Table 4.5: The errors of PUPVMS at *Refinement 1* with different oversampling, $H = 32$, $h = 128$, 32 processors

s	$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{1,\Omega}$	$\|\|p - p^h\|\|_{0,\Omega}$	Local time
0	0.000531367	0.000454854	2.53
1	0.000223261	0.000259422	7.4
2	0.000217938	0.000258164	15.98
3	0.000216099	0.000257963	30.91
4	0.000214664	0.000257889	47.52
5	0.000213509	0.000257848	70.34
6	0.000212574	0.00025782	100.03

Secondly, we will test the parallel efficiency of PUPVMS. The performance of a

Table 4.6: The errors of PUPVMS at *Refinement 2* with different oversampling, $H = 32$, $h = 128$, 32 processors

s	$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{1,\Omega}$	$\|\|p - p^h\|\|_{0,\Omega}$	Local time
0	0.000531367	0.000454854	2.53
1	0.000277955	0.000348831	3.86
2	0.000212182	0.000264358	4.28
3	0.000210285	0.000264576	5.06
4	0.000200008	0.000260023	7.68
5	0.000199918	0.000259997	8.64
6	0.000199911	0.000259983	9.8

parallel algorithm in a homogeneous parallel environment is measured by speedup and parallel efficiency which is commonly calculated by

$$S_p = \frac{T(n_1)}{T(n_2)}, \; E_p = \frac{n_1 \times T(n_1)}{n_2 \times T(n_2)}, \tag{4.26}$$

where $T(n_1)$ and $T(n_2)$ $(n_1 < n_2)$ are the wall time of the parallel program using n_1 and n_2 processor, respectively.

Table 4.7 reports the wall time of PUPVMS at both *Refinement 1* and *Refinement 2* with $h = H/4$ and $s = 1$, the corresponding speedup and parallel efficiency computed with $n_1 = 2$ in (4.26). Figure 4.8, 4.9, 4.10 describe the evolution of the speedup, parallel efficiency and Wall time of PUPVMS with the number of processors respectively. Both Table 4.7 and Figure 4.8, 4.9, 4.10 demonstrate PUPVMS has a good parallel performance. Although, for the speedup and parallel efficiency, PUPVMS at *Refinement 1* is a little better than that of *Refinement 2*, however, from

the view of Wall time, the latter one is better.

Table 4.7: Wall time $T(J)$ in seconds, speedup S_p and parallel efficiency E_p of PUPVMS, $H = 1/32$, $h = 1/128$, $s = 1$.

J	2	4	8	16	32
Refinement1					
$T(J)$	118.95	61.06	32.73	18.29	11.55
$S_p = \frac{T(2)}{T(J)}$	1	1.94808	3.63428	6.50355	10.2987
$E_p = \frac{2\times T(2)}{J\times T(J)}$	1	0.974042	0.90857	0.812944	0.643669
Refinement2					
$T(J)$	65.2	36.66	22.01	13.53	7.69
$S_p = \frac{T(2)}{T(J)}$	1	1.77851	2.96229	4.81892	8.47854
$E_p = \frac{2\times T(2)}{J\times T(J)}$	1	0.889253	0.740572	0.602365	0.529909

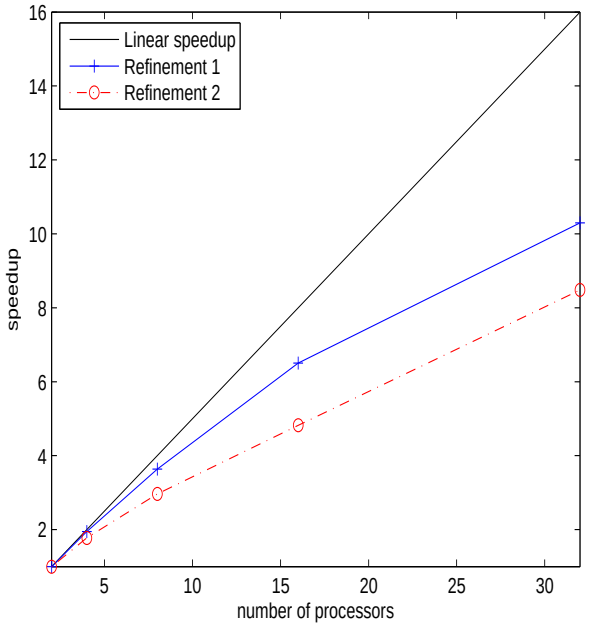

Figure 4.8: The evolution of the speedup of PUPVMS, $H = 1/32$, $h = 1/128$, $s = 1$.

Then, we consider the case of $H = 1/32$, $h = 1/256$, $s = 2$. The corresponding results are shown in Table 4.8 and Figure 4.11, 4.12, 4.13. For this case, we also

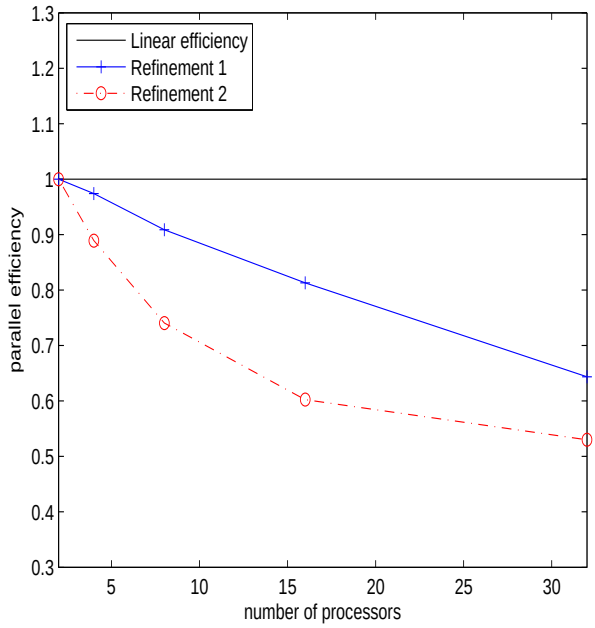

Figure 4.9: The evolution of the parallel efficiency of PUPVMS, $H = 1/32$, $h = 1/128$, $s = 1$.

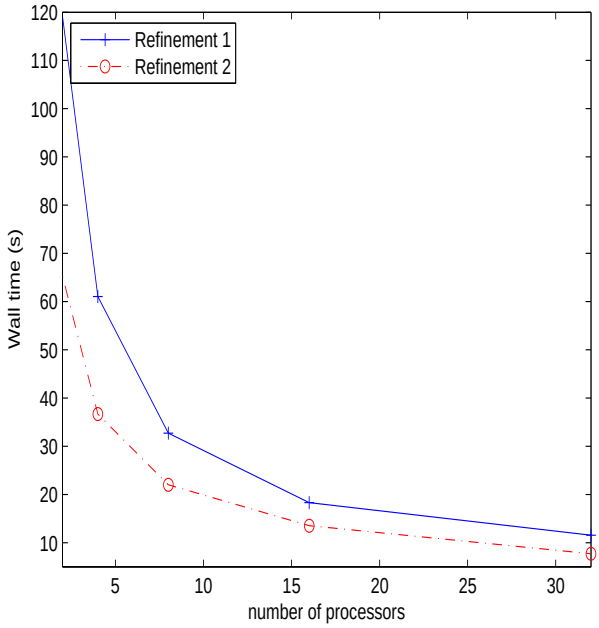

Figure 4.10: The evolution of Wall time of PUPVMS, $H = 1/32$, $h = 1/128$, $s = 1$.

obtain that PUPVMS has a good parallel performance. Moreover, the speedup and parallel efficiency in this case are much better than PUPVMS at the case of $H = 1/32$, $h = 1/128$, $s = 1$, since at this time, the computation size of the sub-problems are larger, and Local time will dominate Wall time, the effect of the cpu time for coarse problem and data communication decrease.

Table 4.8: Wall time $T(J)$ in seconds, speedup S_p and parallel efficiency E_p of PUPVMS, $H = 1/32$, $h = 1/256$, $s = 2$.

J	2	4	8	16	32
Refinement1					
$T(J)$	1302.29	658.12	333.22	171.36	101.53
$S_p = \frac{T(2)}{T(J)}$	1	1.9788	3.9082	7.59973	12.8267
$E_p = \frac{2\times T(2)}{J\times T(J)}$	1	0.989402	0.97705	0.949966	0.801666
Refinement2					
$T(J)$	366.89	187.55	92.35	53.41	33.03
$S_p = \frac{T(2)}{T(J)}$	1	1.95623	3.97282	6.86931	11.1078
$E_p = \frac{2\times T(2)}{J\times T(J)}$	1	0.978113	0.993205	0.858664	0.694236

Finally, we will check the convergence properties of PUPVMS. To get the optimal orders for H^1-norm of velocity and L^2-norm of pressure, we should choose H and h satisfying $h \sim H^{\frac{3}{2}}$. In this example, we compute the finite element solutions by PUPVMS with coarse mesh sizes $H = 1/(16n)$ (n=1, 2, 3) and the corresponding fine mesh sizes $h = H/4$ and $h/8$ for simplicity.

All SFEM, GVMS and Two-GVMS are computed with the same fine mesh sizes on a single processor, see Table 4.9-4.11. For PUPVMS at both *Refinemen-*

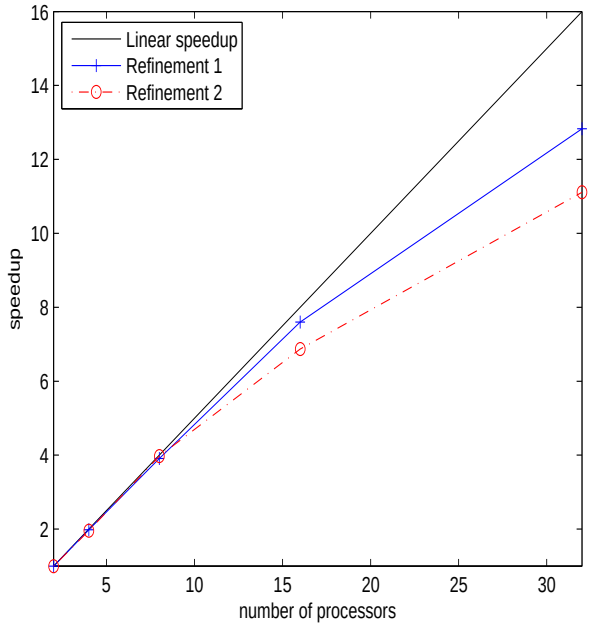

Figure 4.11: The evolution of the speedup of PUPVMS, $H = 1/32$, $h = 1/256$, $s = 2$.

t 1 and *Refinement 2*, the simulations are implemented with 32 processors, for $H = 1/16,\ 1/32,\ 1/48$. For the case of $h = H/4$, we fix $s = 0,\ s = 1$, all computational results are given in Tables 4.12-4.14.

Comparing Table 4.9-4.14, we can see that all three methods reach the optimal convergence orders. The differences are, SFEM, GVMS and Two-GVMS obtain the similar accuracy, and SFEM, GVMS cost the similar CPU time, Two-GVMS are much more efficient than GVMS. Our PUPVMS has the highest efficiency, when $s = 0$, the accuracies are not good enough, which means oversampling are necessary. While $s = 1$, PUPVMS with both *Refinement 1* and *Refinement 2* can yield the approximate solutions with an accuracy comparable to that of SFEM, GVMS and

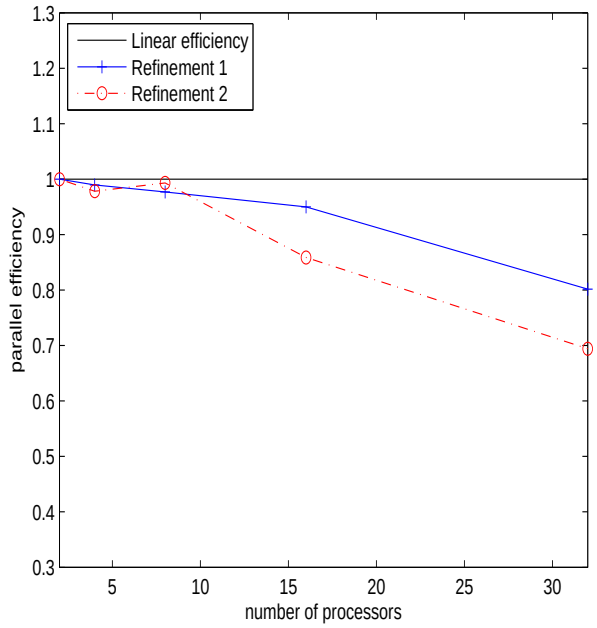

Figure 4.12: The evolution of the parallel efficiency of PUPVMS, $H = 1/32$, $h = 1/256$, $s = 2$.

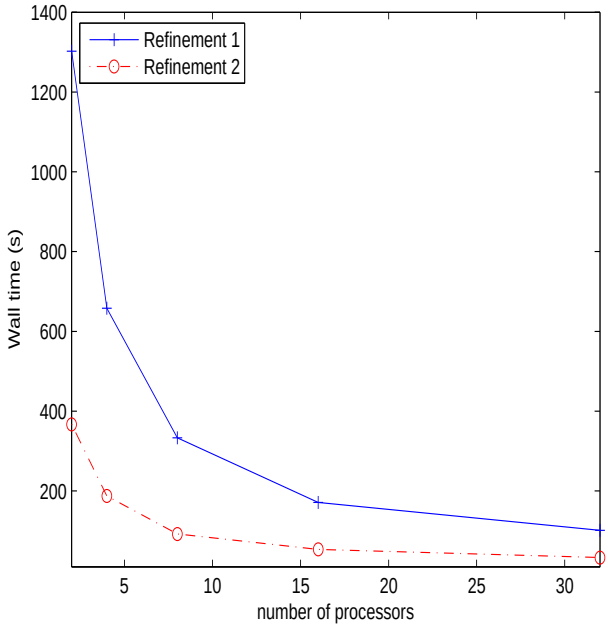

Figure 4.13: The evolution of Wall time of PUPVMS, $H = 1/32$, $h = 1/256$, $s = 2$.

Two-GVMS with the very a little Wall time. PUPVMS with *Refinement 2* is just a little better than that on *Refinement 2*, the advantage is not obvious.

Table 4.9: The errors of SFEM

h	$\|\|\mathbf{u}-\mathbf{u}_h\|\|_{1,\Omega}$	Order	$\|\|p-p_h\|\|_{0,\Omega}$	Order	Wall time
1/64	0.000816817	\	0.00103003	\	8.37
1/128	0.000205013	1.9943	0.00025784	1.99814	41.23
1/192	$9.14038e^{-05}$	1.99225	0.000115185	1.98735	120.54

Table 4.10: The errors of GVMS

h	$\|\|\mathbf{u}-\bar{\mathbf{u}}_h\|\|_{1,\Omega}$	Order	$\|\|p-\bar{p}_h\|\|_{0,\Omega}$	Order	Wall time
1/64	0.000816623	\	0.00103003	\	9.95
1/128	0.000205001	1.99404	0.000257841	1.99813	47.58
1/192	$9.14014e^{-05}$	1.99217	0.000115185	1.98736	133.67

Table 4.11: The errors of Two-GVMS

H	h	$\|\|\mathbf{u}-\bar{\mathbf{u}}_h\|\|_{1,\Omega}$	Order	$\|\|p-\bar{p}_h\|\|_{0,\Omega}$	Order	Wall time
1/16	1/64	0.000816778	\	0.0010299	\	5.81
1/32	1/128	0.00020491	1.99495	0.0002575	1.99986	27.74
1/48	1/192	$9.11727e^{-05}$	1.99725	0.000114446	1.99997	77.73

Table 4.12: The errors of PUPVMS without oversampling, $h = H/4$, $s = 0$

H	h	$\|\|\mathbf{u}-\mathbf{u}^h\|\|_{1,\Omega}$	Order	$\|\|p-p^h\|\|_{0,\Omega}$	Order	Wall time	SwapData time
1/16	1/64	0.00213003	\	0.00186634	\	1.86	0.68
1/32	1/128	0.000531367	2.00309	0.000454854	2.03674	6.12	1.03
1/48	1/192	0.000235956	2.00216	0.000200202	2.02397	14.25	2.39

Table 4.13: The errors of PUPVMS at *Refinement1*, $h = H/4$, $s = 1$

H	h	$\|\|\mathbf{u}-\mathbf{u}^h\|\|_{1,\Omega}$	Order	$\|\|p-p^h\|\|_{0,\Omega}$	Order	Wall time	SwapData time
1/16	1/64	0.000945121	\	0.00104396	\	3.01	0.85
1/32	1/128	0.000223261	2.08177	0.000259422	2.00869	11.55	1.71
1/48	1/192	$9.71595e^{-05}$	2.05193	0.00011531	1.99976	25.74	2.51

Table 4.14: The errors of PUPVMS at *Refinement2*, $h = H/4$, $s = 1$

H	h	$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{1,\Omega}$	Order	$\|\|p - p^h\|\|_{0,\Omega}$	Order	Wall time	SwapData time
1/16	1/64	0.000975335	\	0.00108129	\	2.87	0.55
1/32	1/128	0.000226273	2.10783	0.00026851	2.0171	9.38	1.01
1/48	1/192	$9.98711e^{-05}$	2.00971	0.000119235	2.00212	24.6	2.15

Then, we consider another case, $h = H/8$. In this case, all SFEM, GVMS and Two-GVMS can not work with $h = 1/256,\ 1/384$. When PUPVMS without oversampling ($s = 0$ in Table 4.15), the accuracies are not good enough although it obtains the optimal convergence order. Then, we fix PUPVMS at both *Refinement 1* and *Refinement 2* with two layers oversampling ($s = 2$ in Table 4.16, 4.17), the accuracies are much better, and keep the optimal convergence orders. Compare PUPVMS at *Refinement 1* with that at *Refinement 2*, we find that although they obtain the similar accuracies, and SwapData time are spent nearly the same, however, the latter one cost less Wall time than the former one. In fact, PUPVMS at *Refinement 2* with two layers oversampling cost the similar Wall time and SwapData time with PUPVMS without oversampling, but with much better accuracies.

Table 4.15: The errors of PUPVMS without oversampling, $h = H/8$, $s = 0$

H	h	$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{1,\Omega}$	Order	$\|\|p - p^h\|\|_{0,\Omega}$	Order	Wall time	SwapData time
1/16	1/128	0.0014182	\	0.00116701	\	5.4	1.34
1/32	1/256	0.000350384	2.01705	0.000274489	2.088	27.45	6.92
1/48	1/384	0.000155032	2.01102	0.000119035	2.06058	52.98	15.12

In summary, PUPVMS is stable and has the high efficiency, especially, PUPVMS at *Refinement 2* is much better, and usually, two layers oversampling ($s = 2$) is good

Table 4.16: The errors of PUPVMS at *Refinement1*, $h = H/8$, $s = 2$

H	h	$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{1,\Omega}$	Order	$\|\|p - p^h\|\|_{0,\Omega}$	Order	Wall time	SwapData time
1/16	1/128	0.000320561	\	0.000270283	\	22.75	3
1/32	1/256	$6.92789e^{-05}$	2.21011	$6.6329e^{-05}$	2.02676	101.53	14.61
1/48	1/384	$2.91541e^{-05}$	2.1347	$2.99437e^{-05}$	1.96147	215.99	17.95

Table 4.17: The errors of PUPVMS at *Refinement2*, $h = H/8$, $s = 2$

H	h	$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{1,\Omega}$	Order	$\|\|p - p^h\|\|_{0,\Omega}$	Order	Wall time	SwapData time
1/16	1/128	0.000348141	\	0.000293127	\	7.63	2.73
1/32	1/256	$8.49523e^{-05}$	2.03495	$7.18985e^{-05}$	2.02749	33.03	11.8
1/48	1/384	$3.54307e^{-05}$	2.15681	$3.18908e^{-05}$	2.00495	59.65	16.71

enough for PUPVMS.

To further test our PUPVMS, we also consider another smooth problem in the same domain (referred as problem 2) with exact solution $\mathbf{u} = (u_1, u_2)$,

$$u_1 = \sin(\pi x)^2 \sin(2\pi y),$$

$$u_2 == -\sin(2\pi x) \sin(\pi y)^2,$$

$$p = \cos(\pi x) \cos(\pi y),$$

and $\nu = 1.0$ for simplicity. In present computation, we just use PUPVMS at *Refinement 2*, set the same parameters H, and fix $h = H/8$, 32 processors. The oversampling parameter $s = 2$ is chosen according to the above conclusion. The results are tabulated in Table 4.18. From this table, we can observe the good convergency and high efficient of PUPVMS *Refinement 2* again.

Table 4.18: Problem 2, the errors of PUPVMS at *Refinement 2*, $h = H/8$, $s = 2$

H	h	$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{1,\Omega}$	Order	$\|\|p - p^h\|\|_{0,\Omega}$	Order	Wall time	SwapData time
1/16	1/128	0.00215143	\	0.000691436	\	7.66	2.88
1/32	1/256	0.000516957	2.05718	0.000148109	2.22294	38.01	12.17
1/48	1/384	0.000178728	2.61945	$5.08963e^{-05}$	2.6344	63.3	18.35

The driven cavity flow

A popular benchmark problem for testing numerical schemes is the 'lid driven cavity'. This problem is chosen because some benchmark data is available for comparison. In this problem, computations are carried out in the domain $\Omega = [0, 1] \times [0, 1]$. Flow is driven by the tangential velocity field applied to the top boundary in the absence of other body forces. On the top side $\{(x, 1) : 0 < x < 1\}$, the velocity is equal to $\mathbf{u} = (1, 0)$, and zero Dirichlet conditions are imposed on the rest of the boundary.

Based on the discussion in the above subsection, we will use PUPVMS at *Refinement 2* with two layers oversampling to test this model. Without confusion, we just name PUPVMS for simplicity. Here, we consider the high Reynolds numbers Re =5000, 10000 with fixed $H = 1/64,\ h = 128$, for PUPVMS, we choose $\alpha = 0.1H$ in Step 1 and $\alpha = 0.1h$ in Step 2. The computational results are shown in Figures 4.14-4.15, comparing with the results of Ghia, Ghia, and Shin [31]. Ghia et al.'s algorithm is based on the time dependent stream function using the coupled implicit and multigrid methods. We shall note that with such meshes and Reynolds numbers, SFEM cannot yield a numerical solution because the nonlinear iterations on the

coarse grid fail to converge, and GVMS is stable but asks for a lot of CPU time.

For different Reynolds numbers, the x component of velocity along the vertical centerline and y component of velocity along the horizontal centerlines by PUPVMS are drawn in Fig4.14, 4.15. The accuracy of the computed solutions by PUPVMS have good agreements with the benchmark data of Ghia et al. [31].

Besides, in order to show the stability of PUPVMS, we present the streamlines and the pressure contours of the cavity flows at different Reynolds numbers in Figures 4.16-4.17. As we know, there will appear one main vortex as Reynolds number increases, and it will move to the center of the cavity. Then additional second vortex may appear in the right bottom corner of the cavity, and a third vortex appears in the lower left corner, then the fourth vortex in the up left corner and the fifth smaller vortex in the right bottom corner. Note that, all the streamlines and pressures are constructed by assembling all the local solutions together using the partition of unity functions, which are globally continuous. These results fit to those of Ghia et al. [31]. This test further illustrates the effectiveness of PUPVMS.

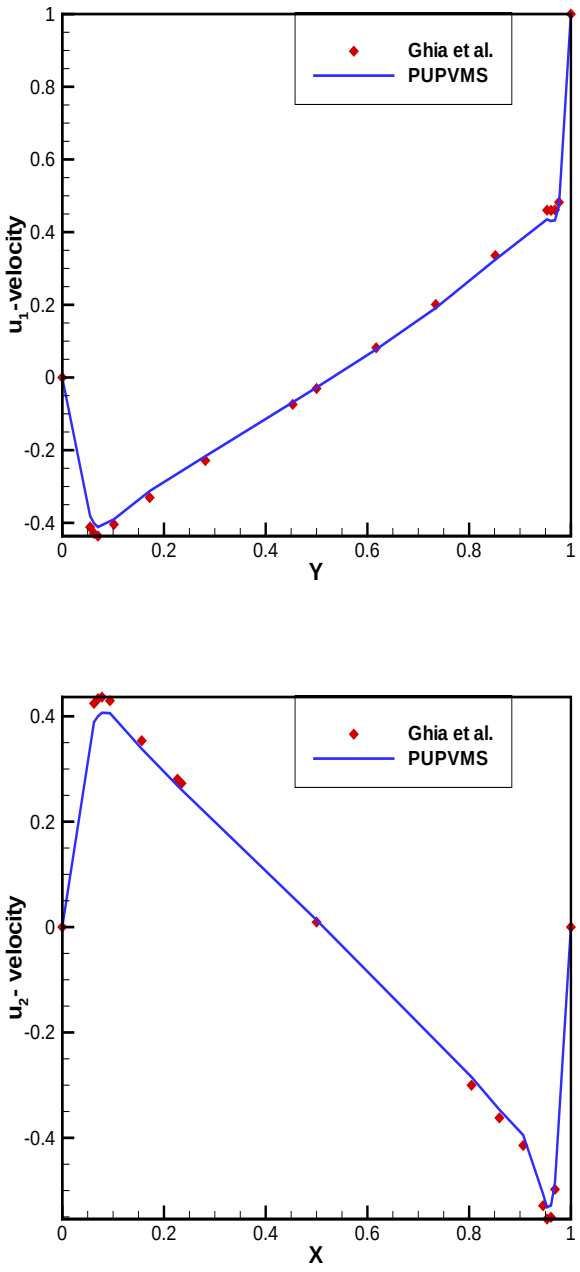

Figure 4.14: PUPVMS for $Re = 5000$. Up, x component of velocity along the vertical centerline; down, y component of velocity along the horizontal centerline

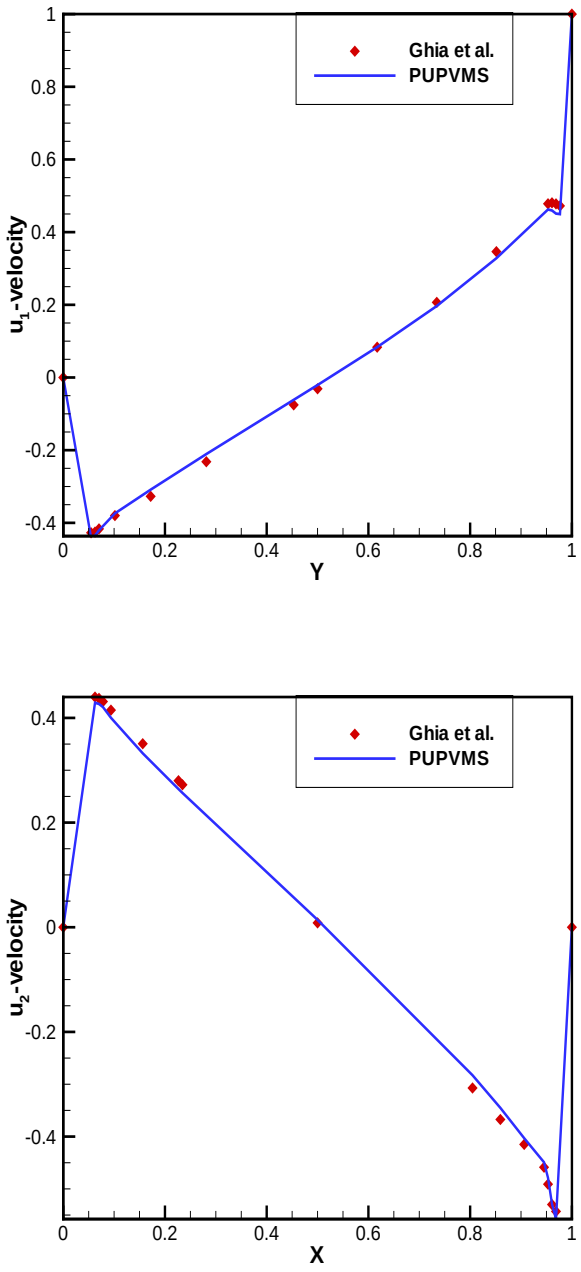

Figure 4.15: PUPVMS for $Re = 10000$. Up, x component of velocity along the vertical centerline; down, y component of velocity along the horizontal centerline.

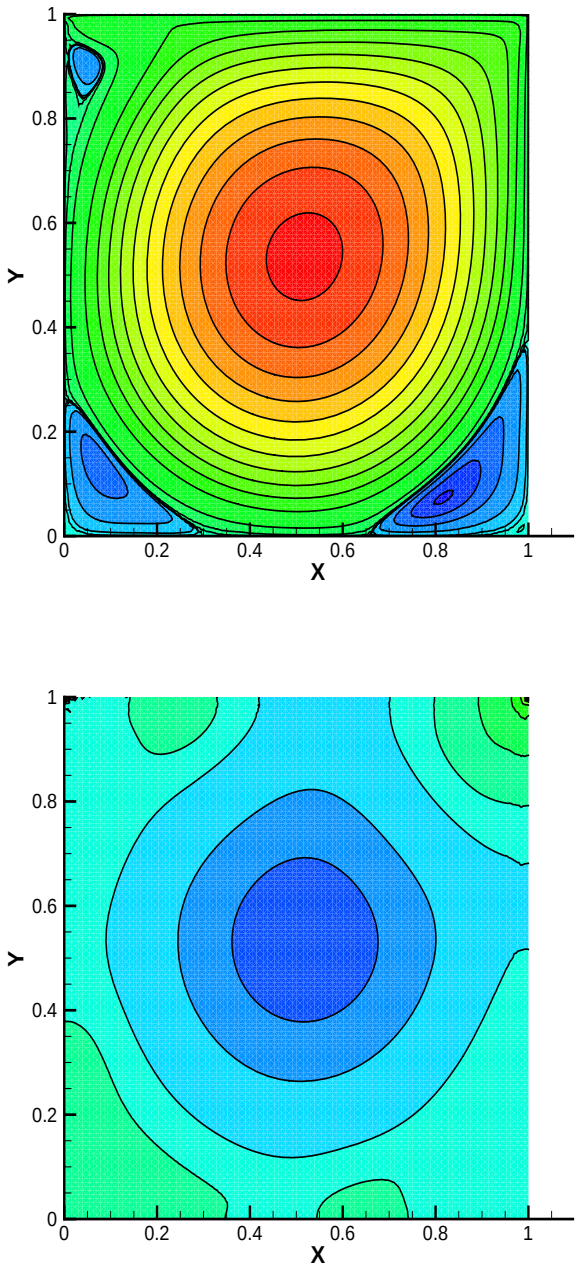

Figure 4.16: PUPVMS for $Re = 5000$. Up, streamlines; down, the pressure contours.

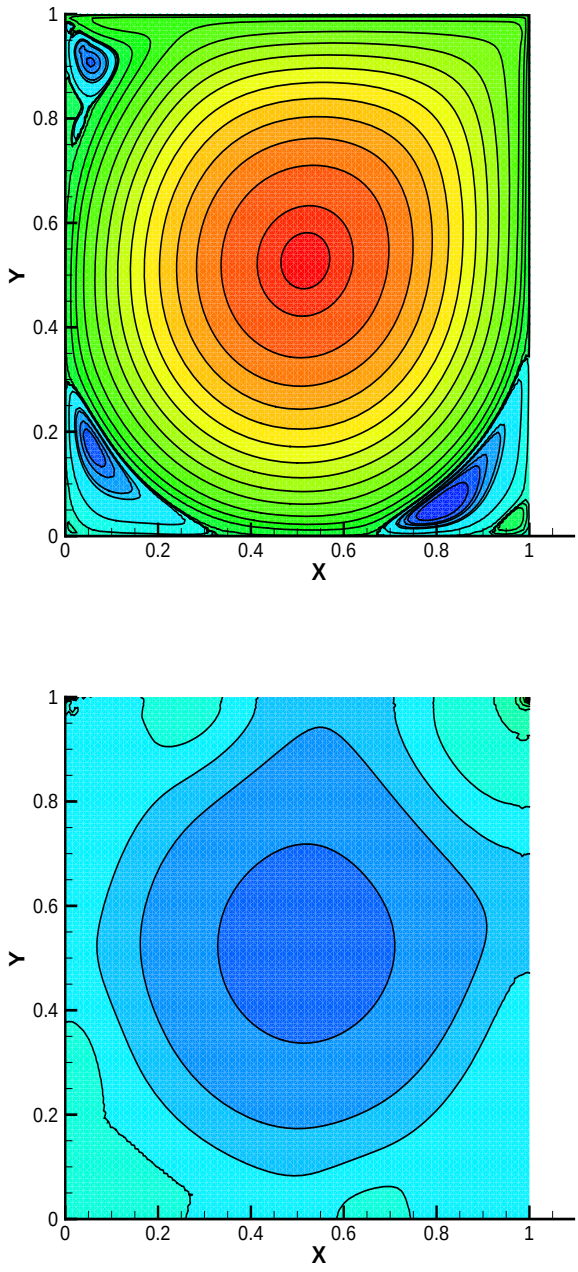

Figure 4.17: PUPVMS for $Re = 10000$. Up, streamlines; down, the pressure contours.

Chapter 5

Conclusions and future work

This chapter draws conclusions on the thesis, and points out some possible research directions related to the work done in this thesis.

5.1 Conclusions

The focus of the thesis has been placed on . Specifically, two research problems have been investigated in detail.

- We propose some a posterior error indicators for the variational multiscale method for the Stokes equations and prove the equivalence between the indicators and the error of the finite element discretization. Some numerical experiments are presented to show their efficiency on constructing adaptive meshes and controlling the error.

- A parallel variational multiscale method based on the partition of unity is proposed for incompressible flows. Based on two-grid method, this algorithm localizes the global residual problem of variational multiscale method into a series of local linearized residual problems. To decrease the undesirable effect of the artificial homogeneous Dirichlet boundary condition of local sub-problems, an oversampling technique is also introduced. The globally continuous finite element solutions are constructed by assembling all local solutions together using the partition of unity functions. Especially, we add an artificial stabilization term in the local and parallel procedure by considering the residual as a subgrid value, which keeps the sub-problems stable. We present the theoretical analysis of the method and numerical simulations demonstrate the high efficiency and flexibility of the new algorithm.

  Another a partition of unity parallel variational multiscale method is proposed. The main difference lies in that in this algorithm we propose two kinds of refinement method. It is difficult to obtain the theoretical result as the above method. However, the numerical simulations show that the error of this algorithm decays exponentially with respect to the oversampling parameter.

5.2 Future work

Related topics for the future research work are listed below.

1. First, we want to apply the method in Chapter 3 to Navier-Stokes equations, especially with large Reynolds number and time-dependent problems. We can also do a posteriori error estimation for other variational multiscale methods, such as those with Bubble functions. Besides, when simulating multiphase flow in porous media we often use the variational multiscale method. Thus we intend to study these problems using this approach.

2. We will try to get the theoretical result of the second method in Chapter 4 thus we can obtain the convergence analysis, which will make the method more convincing. We are also very interested in the extension of PUPVMS to time-dependent problems and more extensive testing for 3D fluid flows.

3. Another aim is to develop some adaptive strategies for oversampling and refinements.

Bibliography

[1] Adams, R. (1975), *Sobolev Spaces*, Academic Press, New York.

[2] Ainsworth, M. and Oden, J. (1993), "A unified approach to a posteriori error estimation using element residual methods," *Numer. Math.*, 65, 23–50.

[3] Ainsworth, M. and Oden, J. (1997), "A posteriori error estimation in finite element analysis," *Comput. Methods Appl. Mech. Engrg.*, 142, 1–88.

[4] Argyris, J. (1954-1955), "Energy theorems and structural analysis, part I: General Theory," *Aircraft Engineering*, 26, 347-356, 383-387, 394; 27, 42-58, 80-94, 125-134, (also published as a book, Butterworths Scientific Publications, London, 1960).

[5] Babuška, I. and Caloz, G.and Osborn, J. (1994), "Special finite element methods for a class of second order elliptic problems with rough coefficients," *SIAM J. Numer. Anal.*, 31, 945–981.

[6] Babuška, I. and Melenk, J. (1997), "The partition of unity method," *Int. J. Numer. Methods Eng.*, 40, 727–758.

[7] Babuška, I. and Miller, A. (1987), "A feedback finite element method with a posteriori error estimation: Part I. The finite element method and some basic properties of the a posteriori error estimator," *Comput. Methods Appl. Mech. Engrg.*, 61, 1–40.

[8] Babuška, I. and Rheinboldt, W. (1978a), "A-posteriori error estimates for the finite element method," *Int. J. Numer. Methods Engrg.*, 12, 1597–1615.

[9] Babuška, I. and Rheinboldt, W. (1978b), "Error Estimates for Adaptive Finite Element Computations," *SIAM J. Numer. Anal.*, 15, 736–754.

[10] Babuška, I. and Vogelius, M. (1984), "Feedback and adaptive finite element solution of one-dimensional boundary value problems," *Numer. Math.*, 44.

[11] Bank, R. and Holst, M. (2000), "A new paradigm for parallel adaptive meshing algorithms," *SIAM J. Sci. Comp.*, 22, 1411–1443.

[12] Bank, R. and Holst, M. (2003), "A new paradigm for parallel adaptive meshing algorithms," *SIAM Rev.*, 45, 291–323.

[13] Belytschko, T. and Black, T. (1999), "Elastic crack growth in finite elements with minimal remeshing," *Int. J. Numer. Methods Engrg.*, 45, 601–620.

[14] Belytschko, T., Moës, N., Usui, S., and Parimi, C. (2001), "Arbitrary discontinuities in finite elements," *Int. J. Numer. Methods Engrg.*, 50, 993–1013.

[15] Benzley, S. (1974), "Representation of singularities with isoparametric finite elements," *J. Numer. Meth. Engrg.*, 8, 537–545.

[16] Byskov, E. (1970), "The calculation of stress intensity factors using finite element with cracked element," *Int. J. Fract. Mech.*, 6, 159–67.

[17] Ciarlet, P. and Lions, J. (1991), *Handbook of Numerical Analysis, Vol. II, Finite Element Methods (Part I)*, North-Holland, Amsterdam.

[18] Clough, R. (1960), "The finile elemenl melhod in plane stress analysis,," in *in Proceedings of the Second ASCE Conference on Electronic Computation*, Pittsburg, Pennsylvania.

[19] Codina, R. and Blasco, J. (2000), "Stabilized finite element method for the transient Navier-Stokes equations based on a pressure gradient projection," *Comput. Methods Appl. Mech. Engrg.*, 182, 277 – 300.

[20] Courant, R. (1943), "Varialional melhods for ihe solulion of problems of equilibrium and vibrations," *Bull. Amer. Math. Soc.*, 49, 1–23.

[21] Daux, C., Moës, N., Dolbow, J., Sukumar, N., and Belytschko, T. (2000), "Arbitrary branched and intersecting cracks with the extended finite element method," *Int. J. Numer. Methods Engrg.*, 48, 1741–1760.

[22] Dolbow, J., Moës, N., and Belytschko, T. (2000), "Discontinuous enrichment in finite elements with a partition of unity method," *Finite Elements in Analysis and Design*, 36, 235–260.

[23] Donea, J. and Huerta, A. (2003), *Finite element methods for flow problems*, Wiley.

[24] Duarte, A. (1996), "The hp cloud method," Ph.d dissertation, University of Texas at Austin, Austin, TX.

[25] Duarte, C. and J., O. (1996), "Hp clouds: A h-p meshless method," *Numer. Methods PDEs*, 12, 673–705.

[26] Duarte, C. and Oden, J. (1996), "An h-p adaptive method using clouds," *Comput. Methods Appl. Mech. Engrg.*, 139, 237–262.

[27] Duarte, C., Kim, D., and Quaresma, D. (2006), "Arbitrarily smooth generalized finite element approximations," *Comput. Methods Appl. Mech. Engrg.*, 196, 33–56.

[28] Fix, G., Gulati, S., and Wakoff, G. (1973), "On the use of singular functions with finite element approximations," *J. Comp. Phys.*, 13, 209–228.

[29] FreeFEM++ (2010), "version 217," *http://www.freefem.org/*.

[30] Fries, T. and Belytschko, T. (2010), "The extended/generalized finite element method: An overview of the method and its applications," *Int. J. Numer. Meth. Engrg.*, 84, 253–304.

[31] Ghia, U., Ghia, K., and Shin, C. (1982), "High-resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method," *J. Comput. Phys.*, 48, 387–411.

[32] Girault, V. and Raviart, P. (1986), *Finite Element Methods for the Navier-Stokes Equations: Theory and Algorithms*, Springer Ser. Comput. Math., 5, Springer, Berlin.

[33] Gravemeier, V., Wall, W., and Ramm, E. (2004), "A three-level finite element method for the instationary incompressible Navier-Stokes equations," *Comput. Methods Appl. Mech. Engrg.*, 193, 1323–1366.

[34] Gunzburger, M. (1989), *Finite Element Methods for Viscous Incompressible Flows: A Guide to Theory, Practice and Algorithms*, Academic Press, Boston.

[35] Hauke, G., Doweidar, M., and Miana, M. (2006), "The multiscale approach to error estimation and adaptivity," *Comput. Methods Appl. Mech. Engrg.*, 195, 1573–1593.

[36] He, Y., Xu, J., and Zhou, A. (2006), "Local and Parallel Finite Element Algorithms for the Navier-Stokes Problem," *J. Comput. Math.*, 24, 227–238.

[37] He, Y., Xu, J., Zhou, A., and Li, J. (2008), "Local and parallel finite element algorithms for the Stokes problem," *Numer. Math.*, 109, 415–434.

[38] Holst, M. (2001), "Adaptive numerical treatment of elliptic systems on manifolds," *Adv. Comput. Math.*, 15, 139–191.

[39] Holst, M. (2002), "Applications of domain decomposition and partition of unity methods in physics and geometry (plenary paper)," *in Proceedings of the Fourteenth International Conference on Domain Decomposition Methods, Mexico City, Mexico. I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds.*

[40] Huang, Y. and Xu, J. (1999), "A partition-of-unity finite element method for elliptic problems with highly oscillating coefficients," in *In: Proceedings for the Work-shop on Scientific Computing*, pp. 27–30, Hong Kong.

[41] Huang, Y. and Xu, J. (2002), "A Conforming Finite Element Method for Overlapping and Nonmatching Grids," *Math. Comput.*, 72, 1057–1066.

[42] Hughes, T. (1995), "Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods," *Comput. Methods Appl. Mech. Engrg.*, 127, 387 – 401.

[43] Hughes, T. and Stewart, J. (1996), "A space-time formulation for multiscale phenomena," *J. Comput. Appl. Math.*, 74, 217–229.

[44] Hughes, T., Feijoo, G., Mazzei, L., and Quincy, J. (1998), "The variational multiscale method - a paradigm for computational mechanics," *Comput. Methods Appl. Mech. Engrg.*, 166, 3–24.

[45] Hughes, T., Mazzei, L., and Jansen, K. (2000), "Large Eddy Simulation and the variational multiscale method," *Comput. Visual. Sci.*, 3, 47–59.

[46] John, V. and Kaya, S. (2005a), "A finite element variational multiscale method for the Navier-Stokes equations," *SIAM J. Sci. Comput.*, 26, 1485–1503.

[47] John, V. and Kaya, S. (2005b), "A finite element variational multiscale method for the Navier-Stokes equations," *SIAM J. Sci. Comput.*, 26, 1485–1503.

[48] John, V. and Kindl, A. (2010), "A variational multiscale method for turbulent flow simulation with adaptive large scale space," *J. of Comput. Phys.*, 229, 301–312.

[49] Klawonn, A. and Pavarino, L. (1998), "Overlapping Schwarz methods for mixed linear elasticity and Stokes problems," *Comput. Meth. Appl. Mech. Engrg.*, 165, 233–245.

[50] Larson, M. and Målqvist, A. (2005), "Adaptive variational multiscale methods based on a posteriori error estimation: Duality techniques for elliptic problems," *Lecture Notes In: Computational Science and Engineering*, 44, 181–193.

[51] Larson, M. and Målqvist, A. (2007), "Adaptive variational multiscale methods based on a posteriori error estimation: Energy norm estimates for elliptic problems," *Comput. Methods Appl. Mech. Engrg.*, 196, 2313–2324.

[52] Li, Y., Mei, L., Li, Y., and Zhao, K. (29), "A two-level variational multiscale method for incompressible flows based on two local Gauss integrations," *Numer. Meth. PDEs.*, 2013, 1986–2003.

[53] Liszka, T., Duarte, C., and Tworzydlo, W. (1996), "H-p meshless cloud method," *Comput. Methods Appl. Mech. Engrg.*, 139, 263–288.

[54] Liu, X. and Li, S. (2006), "A variational multiscale stabilized finite element method for the Stokes flow problem," *Finite Elements in Analysis and Design*, 42, 580–591.

[55] Ma, F., Ma, Y., and Wo, W. (2007), "Local and parallel finite element algorithms based on two-grid discretization for steady NavierCStokes equations," *Appl. Math. Mech.*, 28, 27–35.

[56] Ma, Y., Zhang, Z., and Ren, C. (2006), "Local and parallel finite element algorithms based on two-grid discretization for the stream function form of NavierC Stokes equations," *Appl. Math. Comput.*, 175, 786–813.

[57] Marion, M. and Temam, R. (1998), "Navier-stokes equations: Theory and approximation," in *Numerical Methods for Solids (Part 3) Numerical Methods for Fluids (Part 1)*, vol. 6 of *Handbook of Numerical Analysis*, pp. 503 – 689, Elsevier.

[58] Melenk, J. (1995), "On Generalized Finite Element Methods," Ph.d. thesis, University of Maryland.

[59] Melenk, J. and Babuška, I. (1996), "The partition of unity finite element method: Basic theory and applications," *Comput. Methods Appl. Mech. Engrg.*, 139, 289–314.

[60] Moës, N., Dolbow, J., and Belytschko, T. (1999), "A finite element method for crack growth without remeshing," *Int. J. Numer. Methods Engrg.*, 46, 131–150.

[61] Oden, J., Duarte, C., and Zienkiewicz, O. (1998), "A new cloud-based hp finite element method," *Comput. Methods Appl. Mech. Engrg.*, 153, 117–126.

[62] Pavarino, L. (2000), "Indefinite overlapping Schwarz methods for time-dependent Stokes problems," *Comput. Meth. Appl. Mech. Engrg.*, 187, 35–51.

[63] Rank, E. and Zienkiewicz, O. (1987), "A simple error estimator in the finite element method," *Com. Appl. Numer. Methods*, 3, 243–249.

[64] Shan, L., Layton, W., and Zheng, H. (2013), "Numerical analysis of modular VMS methods with nonlinear eddy viscosity for the Navier-Stokes equations," *Int. J. Numer. Anal. Model.*, 10, 943–971.

[65] Shang, Y. (2011), "A parallel two-level linearization method for incompressible flow problems," *Appl. Math. Lett.*, 24, 364–369.

[66] Shang, Y. (2013), "A parallel two-level finite element variational multiscale method for the Navier-Stokes equations," *Nonli. Anal.-TMA*, 84, 103–116.

[67] Shang, Y. and He, Y. (2010), "Parallel iterative finite element algorithms based on full domain partition for the stationary NavierCStokes equations," *Appl. Numer. Math.*, 60, 719–737.

[68] Shang, Y. and He, Y. (2012), "A parallel Oseen-linearized algorithm for the stationary NavierCStokes equations," *Comput. Methods Appl. Mech. Engrg.*, 209-212, 172–183.

[69] Shang, Y. and Wang, K. (2010), "Local and parallel finite element algorithms based on two-grid discretizations for the transient Stokes equations," *Numer. Algor.*, 54, 195–218.

[70] Shang, Y., He, Y., and Luo, Z. (2011a), "A comparison of three kinds of local and parallel finite element algorithms based on two-grid discretizations for the stationary NavierCStokes equations," *Comput. Fluids*, 40, 249–257.

[71] Shang, Y., He, Y., Kim, D., and Zhou, X. (2011b), "A new parallel finite element algorithm for the stationary Navier-Stokes equations," *Finite Elem. Anal. Des.*, 47, 1262–1279.

[72] Shen, L. (2005), "Parallel Adaptive Finite Element Algorithms for Electronic Structure Computing Based on Density Functional Theory." Ph.D. thesis, Academy of Mathematics and Systems Science, Chinese Acade my of Sciences.

[73] Song, L., Hou, Y., and Zheng, H. (2013), "Adaptive local postprocessing finite element method for the Navier-Stokes equations," *J. Sci. Comput.*, 55, 255–267.

[74] Stewart, J. and Hughes, T. (1997), "An a posteriori error estimator and hp-adaptive strategy for finite element discretizations of the Helmholtz equation in exterior domains," *Finite Elements in Analysis and Design*, 25, 1–26.

[75] Stewart, J. and Hughes, T. (1998), "A tutorial in elementary finite element error analysis: A systematic presentation of a priori and a posteriori error estimates," *Comput. Methods Appl. Mech. Engrg.*, 158, 1–22.

[76] Strouboulis, T., Babuška, I., and Copps, K. (2000a), "The design and analysis of the generalized finite element method," *Comput. Methods Appl. Mech. Engrg.*, 181, 43–69.

[77] Strouboulis, T., Copps, K., and Babuška, I. (2000b), "The generalized finite element method: An example of its implementation and illustration of its performance," *Int. J. Numer. Methods Engrg.*, 47, 1401–1417.

[78] Strouboulis, T., Copps, K., and Babuška, I. (2001), "The generalized finite element method," *Comput. Methods Appl. Mech. Engrg.*, 190, 4081–4193.

[79] Turner, M., Clough, R., Martin, B., and Topp, L. (1956), "Stiffness and deflection analysis of complex structures," *J. Aero. Sci.*, 23, 805–823.

[80] Verfürth, R. (1989), "A posteriori error estimators for the Stokes equations," *Numer. Math.*, 55, 309–325.

[81] Verfürth, R. (1991), "A posteriori error estimators for the Stokes equations II non-conforming discretizations," *Numer. Math.*, 60, 235–249.

[82] Verfürth, R. (1998), "A posteriori error estimates for nonlinear problems. Lr-estimates for finite element discretizations of elliptic equations," *Math. Model. Numer. Anal.*, 32, 817–842.

[83] Xu, J. and Zhou, A. (2000), "Local and Parallel Finite Element Algorithms Based on Two-Grid Discretizations," *Math. Comp.*, 69, 881–909.

[84] Xu, J. and Zhou, A. (2001), "Local and parallel finite element algorithms based on two-grid discretizations for nonlinear problems," *Adv. Comput. Math.*, 14, 293–327.

[85] Xu, J. and Zhou, A. (2002), "Local and parallel finite element algorithms for eigenvalue problems," *Acta Math. Appl. Sin-E.*, 18, 185–200.

[86] Yu, J., Zheng, H., and Shi, F. (2012), "A finite element variational multiscale method for incompressible flows based on the construction of the projection basis functions," *Int. J. Numer. Meth. Fluids*, 70, 793–804.

[87] Yu, J., Shi, F., and Zheng, H. (2014), "Local and parallel finite element algorithm based on the partition of unity for the Stokes problem," *SIAM J. Sci. Comput.*, 36, 547–567.

[88] Zhang, Y., Feng, M., and He, Y. (2010), "Subgrid model for the stationary incompressible Navier-Stokes equations based on the hgih order polynomial interpolation," *Int. J. Numer. Anal. Model.*, 7, 734–748.

[89] Zheng, H., Hou, Y., Shi, F., and Song, L. (2009), "A Finite Element Variational

multiscale method for incompressible flows based on two local Gauss integrations," *J. Comput. Phys.*, 228, 5961–5977.

[90] Zheng, H., Hou, Y., and Shi, F. (2010), "Adaptive Finite Element Variational multiscale method for incompressible flows based on two local Gauss integrations," *J. Comput. Phys.*, 229, 7030–7041.

[91] Zheng, H., Yu, J., and Shi, F. (to appear), "Local and parallel finite element algorithm based on the partition of unity for incompressible flows," *J. Sci. Comput.*

[92] Zienkiewicz, O. and Zhu, J. (1991), "Adaptivity and mesh generation," *Int. J. Numer. Methods Engrg.*, 32, 783–810.