

Application of Projection methods to the Numerical Solution of the Incompressible Navier Stokes Equations

Hongji Zhang

October 2014

A thesis submitted for the degree of Bachelor of Science Honours
Australian National University

Declaration

The work in this thesis is my own except where otherwise stated.

Hongji Zhang

Acknowledgements

The project for this thesis was undertaken at the Department of Mathematics,
The Australian National University.

First I would like to thank my supervisor Steve Roberts for the generous helps
and guidance he has given me.

Second I would like to thank my family and friends who have helped going
through this tough year.

Without those people's support, it would be seemingly impossible for me to
complete the works.

Thank you all

Abstract

This thesis is an investigation of the numerical modelling to the 2D Navier Stokes Equations of Incompressible flow using the well-known “Projection method”. Solving the Navier Stokes equations is never an easy task, a major difficulty lies in the coupling between velocity variables and pressure. The efficiency of solving this coupled equation is not satisfactory especially at the time of last century when the computation power was very limited. Hence the invention of Projection method or (Splitting method) by Alexandre Joel Chorin and Roger Temam independently in the last 1960s was ground breaking in the field of Computational Fluid Dynamics. It basically decouples the velocities and pressure which gives a significant improvement in computation efficiency. In this thesis, we go through the fundamental idea of Projection method which relies on the Helmholtz Hodge Decomposition theorem, Chorin’s original method and all the way to recent second order accurate extensions. In addition, a derivation of Navier Stokes equations is also given in chapter 2.

This thesis is mainly focused on the behaviour of temporal error in projection methods, hence for the sake of simplicity, we have implemented finite difference discretisation scheme. Normal mode analysis was performed in a periodic channel to analyse the order of accuracy of projection methods theoretically. Our analysis shows in periodic channel all proposed second order schemes should show at least second order accuracy in both velocities and pressure, this is also confirmed by our numerical results. In more general domains e.g. Dirichlet boundary conditions, the projection methods however show a degraded accuracy in pressure to about 1.5 order. Only the Gauge method (which is a recent variation of projection methods proposed by E and Liu [6, 14, 33]), shows full second order error convergence in both velocity and pressure. We have also seen the pressure free projection method proposed by Kim and Moin must use a high order approximation to the auxiliary field along the tangential component in order to recover second order

accuracy in pressure. In addition, we have observed an inconsistency in pressure approximation choice with the projection method and a modification was proposed to restore optimal convergence. Interesting flow problems like the Driven cavity problem is also presented for qualitative illustration of the performance of projection methods in case of unsteady, turbulent flows.

Contents

Acknowledgements	v
Abstract	vii
1 Introduction	1
1.1 Motivation	1
1.1.1 Brief history of Navier Stokes Equations	1
1.1.2 Origin of Projection method	3
1.2 Outline of Thesis	3
2 Derivation of Navier Stokes Equations	5
2.1 Continuum hypothesis	6
2.2 Conservation of mass	6
2.3 Balance of Momentum	7
2.3.1 Lagrangian derivative	7
2.3.2 Forces acting on the fluid	8
2.4 Stress tensor for Newtonian fluid	10
2.4.1 Incompressibility	11
3 Numerical Methods	15
3.1 Motivation	15
3.2 Projection method	18
3.3 Helmholtz Hodge decomposition (HHD) Theorem	18
3.4 Application of Helmholtz Hodge Decomposition Theorem to numerical solution of Navier Stokes Equations	23
3.4.1 Chorin's Original projection method	23
3.5 Variations in Projection methods	30
3.5.1 Approximate Projection method	30
3.5.2 Second order accurate schemes	33

3.5.3	Gauge method	39
4	Error Analysis of Projection methods	43
4.1	Introduction of Normal mode analysis	43
4.2	Normal mode analysis of Projection methods	44
4.2.1	Normal analysis for Algorithm 1,2,3	48
4.2.2	Normal mode analysis of Gauge method	65
5	Model implementation	71
5.1	Finite difference scheme	71
5.1.1	Approximation to derivatives	72
5.1.2	Staggered grid	73
5.1.3	Discretisation of Convective and Diffusive terms	75
5.1.4	Linear system solvers	77
5.1.5	update velocity and pressure.	81
6	Numerical Results	83
6.1	Accuracy test	83
6.1.1	Forced flow	84
6.1.2	Unforced flow problem	105
6.2	Driven cavity flow	111
6.2.1	Problem set up	111
6.2.2	Results	112
7	Conclusions	115
	Bibliography	117

Chapter 1

Introduction

In this chapter I will give an introduction to my Thesis, the Navier Stokes equations and the well-popularised Projection method that solves the equations numerically.

1.1 Motivation

Fluid mechanics is the study of the motion of fluid substances, including liquid, gases and plasmas. The motion of fluid substances like in many other physics fields, can be described by a set of mathematical equations with appropriate assumptions imposed. The theory of fluid mechanics rely on the “Continuum hypothesis” which basically treat objects in study as continuous substances. Navier Stokes equations named after Claude-Louis Navier and George Gabriel Stokes is arguably the most fundamental set of equations in fluid mechanics. It is a set of partial differential equations describing the general motion of fluid substances.

1.1.1 Brief history of Navier Stokes Equations

The theoretical development of fluid mechanics date back to the 18th centuries when important works have been published by some of the most influential scientists and mathematicians at the time. Including Leonhard Euler, Jean le Rond d'Alembert, Joseph Louis Lagrange and many more. In 1759 by applying Newton's second law of motion (which basically states the product of mass and acceleration is equal to the sum of body and external forces), Euler was able to provide a set of governing equations that describes fluid motions. Later this was

known to be Euler's equations or Euler's equations of inviscid flow. It describes the motion of fluid neglecting the effect of friction forces arises between surfaces of different fluids or other material in contact. Although mathematically elegant, it has a serious drawback because as was proved by D' Alembert and Euler himself that the drag force acting on a body moving in a steady fluid is always zero. This is in contradiction with real life experimental results. Later this was proven to be the lack of viscosity term in the equations. Hence that's why the theory is only for ideal flows where the viscosity is zero.

In the 19 century, several scientists have worked to fill the gap between theory and experimental observations. In 1822 Claude Louis Marie Henri Navier provided a set of more generalised equations on fluid motions. This was considered to be a major break through, however it was derived by the theory of molecular attraction and repulsion. The equations were re-derived many times by Cauchy (1828), Pisson (1829) and Saint-Venant (1843) each from a rather different point view. Later in 1845 George Gabriel Stokes re-derived again by the introduction of viscosity. More history about the origin of Navier Stokes equations can be found in a beautiful paper by Olivier Darrigol [11].

Although derived from concrete physics principles, Navier Stokes equations at a mathematical level is still not complete. The existence and uniqueness of smooth solutions in 3-D is yet to be proven (or disproven). This is really a bottleneck for the theory. At the turn of the 20th century, especially through the development of computer digital power, the numerical solutions of Navier Stokes equations seem to provide a viable alternative. In fact, one of the founding fathers of Computational Fluid Dynamics (CFD), John Von Neumann has predicated that the computer simulated solutions would eventually replace analytical solutions and make the laboratory experiments obsolete. Indeed his predication did not come true completely, nevertheless the study of computer generated numerical solutions do become increasingly important.

The computation of numerical solutions for Navier Stokes equations are actually not very straightforward to do so because the pressure and velocities are coupled with each other. Solving velocities and pressure individually requires the knowledge of each other. Thus they have to be solved together. Not only this brings heavy computational effort, but also the linear system resulted leads to a "saddle" point problem. Especially in the first half of last century when the

computing power was much less powerful as what it stands today. Alternative ways to compute the numerical solutions were strongly needed at the time.

1.1.2 Origin of Projection method

Alexandre Joel Chorin and Roger Temam in late 1960s have independently proposed a very clever alternative method which now known as “Projection method”. It basically decouples the pressure and velocities so that each variable can be solved independently, thus avoid the difficulties the coupled equations have. This was proven to be a very robust and efficient method. To date, this is still the popular method which most modern Navier Stokes solvers rely on. However the improvement in efficiency comes at a price. The accuracy of Projection methods is often questioned, especially for the pressure variable. The accuracy of velocity variable can be easily improved whereas pressure is not. In fact many researchers have argued that the numerical pressure can only be first order accurate in time. Although many recent papers have shown second accurate schemes, however this is still an open question. Some papers even show slight contradictory results in the literature [6, 14, 15, 24]. In this thesis, we examines the second order accurate extensions of Chorin’s original projection methods and provide theoretical and numerical investigations on their overall accuracy.

1.2 Outline of Thesis

Here I will give a basic outline of my thesis. The first chapter is an introduction to Navier Stokes equations and Projection method which is widely used in computing numerical solutions. The second chapter presents the derivation of Navier Stokes equations from first physics principles. In chapter 3 we first talk about the limitation of the coupled solver and then give a detailed explanation of Projection methods, including its original idea from the theory of Helmholtz Hodge decomposition; application of the theorem to the formation of Chorin’s original projection method and recent second order modifications. In chapter 4 we give a brief analysis on the accuracy of projection methods through normal mode analysis whereas chapter 5 talks about the numerical implementation of our Navier Stokes solver; finally we present our results about the performance of projection methods through numerical validations in chapter 6. Conclusions about accuracy of projection methods are then made in chapter 7.

Chapter 2

Derivation of Navier Stokes Equations

There are many ways described in literature in which Navier Stokes can be derived. In fact, during the earlier years of the theory establishment, the equations were re-derived many times using different methods, such as the molecular attraction and repulsion argument used by Navier. There are also different forms the equations can take. For instance the incompressible and compressible forms and stream function formulation. In this chapter, we present a simple direct method to derive the 3-D Incompressible Navier Stokes equations in Cartesian coordinates via first physics principles.

Problem set up We want to develop a mathematical model which describes the motion of a fluid in a region of space.

We use a standard Euclidean coordinates and in 3D. Let Ω denote the spatial domain filled with a fluid such as water (or gases). Further we take Ω as a bounded subset of \mathbb{R}^3 . We are focused on the motion of an arbitrary fluid particle in Ω . Let $\mathbf{x} \in \Omega$ be a vector field that describes the location of the fluid particle. In 3D, $\mathbf{x} = (x(t), y(t), z(t), t)$. Let $u(\mathbf{x}(t), t) = (u(x, y, z, t), v(x, y, z, t), w(x, y, z, t))$ be the spatial velocity of the fluid which is the time derivative of $\mathbf{x}(t)$. Our mathematical model depends on several basic assumptions and physics principals.

2.1 Continuum hypothesis

This is the fundamental assumption we assume and it basically means that we are considering the fluid as continuous matter consisted of infinitesimal points. Strictly speaking this assumption is obviously "Wrong" as the fluid is not continuous but rather made up of discrete molecules. They are not infinitesimal points but rather have physical dimension that can be measured. Nevertheless the continuum hypothesis does lead to very accurate results for most macroscopic phenomena [9].

It follows from the hypothesis that we assume the functions such as u and density ρ are smooth too. This means for any sub-region W of Ω , the fluid has a well-defined density:

$$m(W, t) = \int_W \rho(\mathbf{x}, t) dV \quad (2.1)$$

where dV is the volume of W . We also denote W as the control volume.

However this is not enough to derive the equation of motion of fluid particles. We need further two conservation laws: Conservation of mass; Balance of momentum (Newton's second law of motion) and Conservation of energy [9]. We will consider these fundamental principles one by one throughout the derivation. In fact in the case of incompressible flow equations which is what we are aiming to achieve, the conservation of energy is replaced by the incompressibility constraint as pressure is not a thermodynamic variable in this case.

2.2 Conservation of mass

This is one of the most and earliest fundamental laws in classical physics. It states that the mass of a closed system must remain constant over time. Hence in the context of our derivation, this means that in the control volume W , the rate of change of mass inside is equal to the rate of which mass is crossing the boundary ∂W in the inward direction. This is formally expressed in the following equation:

$$\frac{d}{dt} m(W, t) = \frac{d}{dt} \int_W \rho(\mathbf{x}, t) dV = - \int_{\partial W} \rho \mathbf{u} \cdot \mathbf{n} dA \quad (2.2)$$

where \mathbf{n} is the surface normal to the boundary ∂W (defined to take the outward direction) and A is the area of the interface along the boundary between W and the rest of D .

Because both density and its partial derivatives are assumed to be continuous and hence we can use Leibniz's rule to bring the differentiation into an integral:

$$\int_W \frac{\partial \rho}{\partial t}(\mathbf{x}, t) dV = - \int_{\partial W} \rho \mathbf{u} \cdot \mathbf{n} dA \quad (2.3)$$

This is called the "Integral form" of the law of conservation of mass.

By applying the divergence theorem we arrive at the "Differential form" of the law of conservation of mass:

$$\int_W \frac{\partial \rho}{\partial t}(\mathbf{x}, t) dV + \int_W \nabla(\rho \mathbf{u}) dV = 0 \quad (2.4a)$$

\Rightarrow

$$\frac{\partial \rho}{\partial t}(\mathbf{x}, t) + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.4b)$$

Where $\nabla \cdot = \sum_{i=1}^3 \frac{\partial}{\partial x_i}$ is the divergence operator in a 3-dimensional vector space. This equation is also known as the "Continuity Equation". Differential form is very important in the subsequent derivations and analysis. However in case the density and velocity do not have continuous partial derivatives then we would need to use the Integral form of the conservation law [7].

2.3 Balance of Momentum

2.3.1 Lagrangian derivative

In our spatial domain Ω , recall the path followed by a fluid particle is: $\mathbf{x} = (x(t), y(t), z(t))$. Hence the path or trajectory followed by the particle is varying with respect to time. The velocity of such a particle at a particular location and at an instant time is given by $\frac{d\mathbf{x}}{dt} = \mathbf{u}(x(t), y(t), z(t))$ which is just an ordinary derivative. The question now comes as how should we measure the change in velocity? This introduces the concept of acceleration: $\mathbf{a}(t) = \frac{d}{dt}\mathbf{u}$ which is merely the time derivative of the velocity. An intrinsic approach to find $\mathbf{a}(t)$ is using the standard Eulerian derivative which is simply written as $\partial_t \mathbf{u} = \frac{\partial \mathbf{u}}{\partial t}$. This is indeed

a valid approach but only when we are measuring the velocity at a fixed point. Nevertheless, in our case because of the (space and time dependent) velocity field presented in the region, the fluid particles are constantly being transported throughout the domain Ω . As a result, using the Eulerian approach which sets the reference point fixed in space does not help us to track down the path of a fluid particle. Fortunately the Lagrangian approach solves this problem because it sets the reference point to be moving along with the fluid particle. Hence this allows tracking of the particle possible. This might sounds difficult to formulate but we can actually express the Lagrangian derivative in terms of the Eulerian derivative. By using Chain rule to the acceleration of a fluid particle we get:

$$\begin{aligned}\mathbf{a}(t) &= \frac{d}{dt} \mathbf{u}(x(t), y(t), z(t)) \\ &= \frac{\partial \mathbf{u}}{\partial x} \frac{dx}{dt} + \frac{\partial \mathbf{u}}{\partial y} \frac{dy}{dt} + \frac{\partial \mathbf{u}}{\partial z} \frac{dz}{dt} + \frac{\partial \mathbf{u}}{\partial t}\end{aligned}\tag{2.5a}$$

where $\hat{x}, \hat{y}, \hat{z}$ are unit directional vectors in the x, y and z directions respectively
Rewriting this equation in a concise way we obtained:

$$\mathbf{a}(t) = \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u}\tag{2.5b}$$

where we have used the standard “subscript” notation to indicate derivatives (i.e \mathbf{u}_t means the time derivative of velocity \mathbf{u})

The Lagrangian derivative (also referred as the material derivative) for any space-time dependent function (scalar or vector) can therefore be defined as [7]

$$\frac{D}{Dt} = \partial_t + \mathbf{u} \cdot \nabla\tag{2.6}$$

2.3.2 Forces acting on the fluid

To fully understand how the fluid motion in Ω is governed, we need to understand the forces acting on the fluid substances. Let's consider the control volume W again. There are two classes of forces acting on fluid in W : Surface forces and Body forces.

Let's consider the surface forces first. It is the force acting across the surface S of the control volume by the rest of the fluid. The stress which is the force per unit area can be broken down into the normal stress (acting normally to the surface S) and viscous stress (acting tangentially to the surface). The normal stress is related to the concept of pressure which is represented by a scalar field

$p(\mathbf{x}, t)$. The viscous stress tensor (σ) describes the viscous force. In 3D, it is just a matrix. In an ideal flow (Euler's inviscid flow) only the normal stress is considered, which neglects the transport of momentum across the surface S between the control volume and the rest of the fluid. This is clearly non-realistic and thus this is why it is coined as ideal or perfect flow. Hence in our derivation of the more general Navier Stokes equations, the viscous stress must be included. Thus the total surface force (per unit area) is written as:

$$-p(\mathbf{x}, t)\mathbf{n} + \sigma(\mathbf{x}, t) \cdot \mathbf{n} \quad (2.7)$$

Now we can integrate this to find the total surface force across ∂W by the use of divergence theorem:

$$\mathbf{S}_{\partial W} = - \int_{\partial W} p \mathbf{n} \, dA + \int_{\partial W} \sigma \cdot \mathbf{n} \, dA \quad (2.8a)$$

where A is the area of the surface S .

Consider the two integrals separately: for pressure:

If \mathbf{e} is any fixed unit vector in space, then

$$\begin{aligned} -\mathbf{e} \cdot \int_{\partial W} p \mathbf{n} &= - \int_{\partial W} p(\mathbf{e} \cdot \mathbf{n}) \, dA \\ &= - \int_W \nabla \cdot (p\mathbf{e}) \, dV, \quad \text{by the divergence theorem} \\ &= \mathbf{e} \cdot - \int_W \nabla p \, dV \end{aligned} \quad (2.8b)$$

by dropping the unit vector \mathbf{e} we obtain the total surface force as

$$- \int_{\partial W} p \mathbf{n} \, dA = - \int_W \nabla p \, dV \quad (2.8c)$$

For the viscous stress: By using divergence theorem

$$\int_{\partial W} \sigma \cdot \mathbf{n} \, dA = \int_W \nabla \cdot \sigma \, dV \quad (2.8d)$$

Hence the total surface force is:

$$\mathbf{S}_{\partial W} = - \int_W \nabla p \, dV + \int_W \nabla \cdot \sigma \, dV \quad (2.8e)$$

Now if there is also a body force applied to the fluid, usually it is the gravity. Sometimes it can also be an external forcing term. Assume the body force F has density \mathbf{f} then:

$$\mathbf{F} = \int_W \rho \mathbf{f} \, dV \quad (2.9)$$

Combine the surface and body force densities we obtain the total force density:

$$\text{force (per unit volume)} = -\nabla p + \nabla \cdot \sigma + \mathbf{f} \quad (2.10)$$

Hence by Newton's second law of motion which is the conservation momentum we obtained:

$$\rho \frac{D\mathbf{t}}{Dt} = -\nabla p + \nabla \cdot \sigma + \rho \mathbf{f} \quad (2.11a)$$

where D is the Lagrangian (material) derivative.

Or if we expand the Lagrangian derivative:

$$\rho \frac{\partial \mathbf{u}}{\partial t} = -\rho(\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla p + \nabla \cdot \sigma + \rho \mathbf{f} \quad (2.11b)$$

This is the differential form of the conservation of momentum.

2.4 Stress tensor for Newtonian fluid

The stress tensor σ is closely related to the rate of strain or deformation. In fact for Newtonian flow σ is a linear function of rate of strain.

First we demonstrate that the velocity field of the fluid can be separated into sum of (rigid) translation, a deformation and (rigid) rotation.

If we consider the fluid particle initially at position \mathbf{x} in \mathbb{R}^3 , then let it move a small distance h along the path \mathbf{h} (so h is the length of the vector \mathbf{h}). We write the velocity field at point $(\mathbf{x} + \mathbf{h})$ as:

$$\mathbf{u}(\mathbf{x} + \mathbf{h}) = \mathbf{u}(\mathbf{x}) + \mathbf{D}(\mathbf{x}) \cdot \mathbf{h} + \frac{1}{2} \xi(\mathbf{x}) \times \mathbf{h} + \mathcal{O}(h^2) \quad (2.12)$$

where $\mathbf{D}(\mathbf{x})$ is called the deformation tensor, in 3D, it is a symmetric matrix; ξ is called the rotation vector.

To find $\mathbf{D}(\mathbf{x})$ and $\xi(\mathbf{x})$ we can Taylor expand the above expression for $\mathbf{u}(\mathbf{x} + \mathbf{h})$ at point \mathbf{x} . This lead to

$$\mathbf{u}(\mathbf{x} + \mathbf{h}) = \mathbf{u}(\mathbf{x}) = \nabla \mathbf{u}(\mathbf{x}) \cdot \mathbf{h} + \mathcal{O}(h^2) \quad (2.13)$$

In 3D, we can separate the matrix $\nabla \mathbf{u}(\mathbf{x})$ as one symmetric and antisymmetric part by simple algebra. Define:

$$\mathbf{D} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T] \quad (2.14a)$$

$$\mathbf{S} = \frac{1}{2}[\nabla \mathbf{u} - (\nabla \mathbf{u})^T] \quad (2.14b)$$

so that $\nabla \mathbf{u} = \mathbf{D} + \mathbf{S}$.

We are more interested in the deformation matrix \mathbf{D} which formally is:

$$\begin{bmatrix} 2\partial_x u & \partial_y u + \partial_x v & \partial_z u + \partial_x w \\ \partial_x v + \partial_y u & 2\partial_y v & \partial_z v + \partial_y w \\ \partial_x w + \partial_z u & \partial_y w + \partial_z v & 2\partial_z w \end{bmatrix}$$

For Newtonian flows we have the nice property that the stress tensor σ is a linear function of \mathbf{D} . In fact we can write σ as:

$$\sigma = 2\mu\mathbf{D} + \lambda\nabla \cdot \mathbf{u} \quad (2.15)$$

where we introduce the terms: μ is the first coefficient of viscosity and λ the second coefficient of viscosity [9]. Here we only give a basic introduction to stress tensors, more details can be found in textbooks in fluid dynamics, for instance the well-cited book by Chorin [9]

Substitute this expression of σ back into our balance of momentum equation (the part with $\nabla \cdot \sigma$) and after some algebra we recover the Navier Stokes equations.

In a compact form:

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + (\lambda + \mu)\nabla(\nabla \cdot \mathbf{u}) + \mu\nabla^2\mathbf{u} + \rho\mathbf{f} \quad (2.16)$$

2.4.1 Incompressibility

Now we have 4 equations (balance of momentum (for u, v, w and continuity equation) corresponds to 5 unknowns, namely the velocities: u, v, w , pressure p and density ρ). Obviously these are not enough to provide a complete solutions to our fluid flow problem. This is when the conservation of energy comes into play. However the derivation of conservation of energy requires advanced thermodynamics and we exclude the details here. We need to find another constraint to replace it. Because we are working with incompressible flows, this implies that the fluid actually cannot be compressed or stretched. Hence this leads to the fact

that the density is constant over time ($\frac{\partial \rho}{\partial t} = 0$). If we go back to the continuity equation this results in:

$$\begin{aligned}\frac{D\rho}{Dt} &= \nabla \cdot (\rho \mathbf{u}) \\ &= 0\end{aligned}\tag{2.17a}$$

if further assume that the density is also constant in space then we can drop ρ in the previous equation and obtain

$$\nabla \cdot \mathbf{u} = 0\tag{2.17b}$$

This is the well-known divergence free constraint. This is the additional constraint we need to complete the derivation. Now we have 4 equations (balance of momentum and incompressibility constraint) for 4 unknowns: velocities and pressure. The density drops out.

If we add this divergence free constraint into the our Navier Stokes equations in the previous subsection we find the momentum equation greatly simplifies:

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}\tag{2.18}$$

Expand the material derivative we finally recover the set of Navier Stokes equations for Incompressible Newtonian flow::

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{cases}\tag{2.19}$$

The part $(\mathbf{u} \cdot \nabla) \mathbf{u}$ is called the advective inertial force or convection term whereas $\nabla^2 \mathbf{u}$ is referred to viscous force.

Non-dimensionalisation In practice, fluid motion is best analysed in the non-dimensionalised Navier Stokes form. This is done by dividing the physical variables: velocities, time and displacement of a flow by their characteristic scales so the these variables are now dimensionless. The choice of characteristic scales vary between flow problems and depends on the geometry of the flow too. For instance, for flow passing a sphere, the characteristic length L is usually taken to be the diameter of the sphere and the characteristic velocity U is taken to be the velocity at infinity. Time also has a characteristic scale which is taken to be the ratio between U and L . [9]

Hence by non-dimensionalisation

$$\mathbf{u}' = \frac{\mathbf{u}}{U}, \quad \mathbf{x}' = \frac{\mathbf{x}}{L}, \quad t' = \frac{t}{T} \quad (2.20)$$

If we substitute these back into the Navier Stokes equations we obtain the non-dimensionalised Navier Stokes equations:

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{R} \nabla^2 \mathbf{u} + \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (2.21)$$

where we have defined $R = \frac{LU}{\nu}$ to be the Reynolds number. The magnitude of parameter Reynolds number describes the ratio between advective inertial force and viscous force. This leads to the important concept of “Dynamic similarity” which states two flows with the same geometry and Reynolds number would result in similar flow patterns. This is very important in fluid dynamics analysis. Therefore the non-dimensionalised Navier Stokes equations are more commonly used. In our subsequent numerical analysis we will be using the non-dimensionalised form too.

Chapter 3

Numerical Methods

3.1 Motivation

In this chapter we investigate the numerical solvers to the Incompressible Navier Stokes equations based on the Projection methods.

We have 2 types of variables to solve for: velocities (\mathbf{u}) and pressure p . Intuitively we are looking for numerical schemes which given the velocities and pressure at the initial time step with boundary conditions, computes the velocity and pressure at a later time.

Let us consider the Navier Stokes equations in spatial domain $\Omega \subset \mathbb{R}^2$ and time interval: $[0, T]$, $T > 0$ with given initial value and boundary conditions.

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{R} \nabla^2 \mathbf{u} \\ \nabla \cdot \mathbf{u} = 0 \\ \mathbf{u}(x, y, t = 0) = \mathbf{u}_0 \\ \mathbf{u}(x, y, t) = \mathbf{u}_b \quad \text{with boundary points: } x, y \in \partial\Omega \end{cases} \quad (3.1)$$

Here \mathbf{u} with subscript b denotes the boundary velocity values specified by the problem.

We take $\mathbf{u}(\mathbf{x}, t)$ to be the smooth analytic solution whereas \mathbf{u}^n with the superscript n to denote the numerical solution discretised at time iteration n . This is corresponding to the time at $t = \Delta t n$ with $n = 0, 1, \dots, N$ where N is the maximum iteration. The time stepping is determined by the ratio: $\frac{T}{N}$. Same

notation holds for pressure.

Let us first consider a first order backwards Euler discretisation in time to the above initial value problem: assume we have the knowledge of \mathbf{u}^n and p^n

$$\begin{cases} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla_h p^{n+1} + (\mathbf{u}^n \cdot \nabla_h) \mathbf{u}^n + \frac{1}{R} \nabla_h^2 \mathbf{u}^{n+1} + f^n \\ \nabla_h \cdot \mathbf{u}^{n+1} = 0 \end{cases} \quad (3.2)$$

where the subscript “ h ” refers to numerically discretised operators (e.g. $\nabla_h \cdot$ means the numerical divergence operator).

This is an implicit scheme in time for the viscous term. We have chosen this over explicit schemes because it is usually numerical more stable.

The above equations can be rearranged into a block matrix form for convenience:

$$\begin{bmatrix} A & \Delta t \nabla_h \\ -\Delta t \nabla_h \cdot & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (3.3)$$

Where for simplicity we have defined $A = I - \frac{\Delta t}{R} \nabla_h^2$ (I stands for the identity matrix) and $F = \mathbf{u}^n - \Delta t (-(\mathbf{u}^n \cdot \nabla_h) \mathbf{u}^n + f^n)$

Assume the discrete divergence and gradient operators are skew adjoint i.e. $-\nabla_h \cdot = (\nabla_h)^T$, hence by denoting $D = -\nabla_h \cdot$ we obtained a symmetric block matrix:

$$\begin{bmatrix} A & \Delta t \nabla_h \\ (\Delta t \nabla_h)^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (3.4)$$

This block matrix is actually indefinite in the sense that it has both positive and negative eigenvalues. Hence this leads to a saddle point problem when solving it. This causes many trouble in practice.

First if we consider this as an optimisation problem, then such a saddle point problem makes it hard to find the minimiser (\mathbf{u}^{n+1}) . We will illustrate this in the following simple analysis.

The previous numerical scheme is equivalent to the constraint optimisation problem.

Essentially we want to minimise the energy

$$J = \frac{1}{2} \mathbf{u}^{n+1 T} A \mathbf{u}^{n+1} - F^T \mathbf{u}^{n+1}$$

subject to the divergence free constraint:

$$-\Delta t \nabla_h \cdot \mathbf{u}^{n+1} = 0$$

The optimisation problem is tackled by the common Lagrangian approach. The Lagrangian function is defined as:

$$\Lambda = \frac{1}{2} \mathbf{u}^{n+1T} A \mathbf{u}^{n+1} - F^T \mathbf{u}^{n+1} - \Delta t \lambda \nabla_h \cdot \mathbf{u}^{n+1}$$

where $\lambda \in \mathbb{R}$ is called the Lagrange multiplier.

The solutions (or minimisers) are obtained by taking the gradient of Λ (differentiate with respect to \mathbf{u}^{n+1})

$$\nabla_h \Lambda = A \mathbf{u}^{n+1} - F + \Delta t \nabla_h \lambda = 0$$

However by solving for this equation we will find that the critical points obtained are saddle points.

Interestingly if we combine the above equation with the divergence free constraint we recover a block matrix similar to what we considered before for the numerical scheme. Now we find that the velocity variable presents in both formulations, but the pressure is replaced by the Lagrange multiplier in the optimisation formulation.

$$\begin{bmatrix} A & \Delta t \nabla_h \\ \Delta t \nabla_h^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (3.5)$$

Therefore from the numerical optimisation point of view we see that in incompressible flows, the pressure is merely a mathematical constant added to preserve the divergence free constraint. Hence it is not a thermodynamic variable [21, 23]

Second we illustrate the coupled numerical scheme presented is computationally very expensive to solve. We try to solve it by performing Gaussian elimination to the block matrix in Equation 3.4.

Then we see that the velocity and pressure are solved together in two steps:

$$\mathbf{u}^{n+1} = A^{-1} (F - \nabla_h (\nabla_h^T A^{-1} \nabla_h)^{-1} (\nabla_h^T A^{-1} F)) \quad (3.6a)$$

and

$$p^{n+1} = (\nabla_h^T A^{-1} \nabla_h)^{-1} (\nabla_h^T A^{-1} F) \quad (3.6b)$$

Since A contains a Laplace term and so the method involves calculating Laplacian and its inverses many times in each time iteration which clearly indicates a huge computational costs! We really want to find alternative ways to solve for \mathbf{u}^{n+1} and p^{n+1} . This leads to the development of Projection methods.

3.2 Projection method

In the first half of last century, when the computer digital power was not as strong as it stands today, the coupled solver do give raises serious issue in lack of efficiency. Therefore some researchers in the late 1960s including Chorin and Temam started to think methods to decouple velocity and pressure. The idea is to split the relation between these two quantities and solving them individually (or successively). For instance, solve for velocity at iteration $n + 1$ and then correct it by imposing the divergence free constraint. The pressure can then be solved accordingly. The question now is how to find a way to correct the divergence of velocity? Both Chorin and Temam turned to the Helmholtz Hodge Decomposition theorem and used it as a theoretical foundation for their numerical methods [6, 7, 9, 21, 29]. Late this type of decoupling method was coined to be the famous *Projection Method* or *Splitting method* or sometimes *Fractional Step Method* [6, 18]

In this section we talk about the original idea of projection method and how it is used to solve the Navier Stokes equations.

3.3 Helmholtz Hodge decomposition (HHD) Theorem

The Helmholtz theorem or Helmholtz Hodge decomposition theorem is a fundamental theorem in vector calculus. It basically states that for any smooth vector field in the subset of \mathbb{R}^2 or \mathbb{R}^3 , it can be uniquely decomposed into the sum of irrotational (curl free) and solenoid (divergence free) vector fields [3, 9]. For the sake of simplicity, the complete proof of the theorem is not included, but we rather focus on the application of the theorem to projection methods. Nevertheless, we will show the important results of the theorem, namely the orthogonality, existence and uniqueness of the decomposition. More detailed description of the theorem can be found in many textbooks and papers, for instance the book by

Arfken [3].

Theorem 3.1. *Let Ω be a bounded regular domain of \mathbb{R}^3 , let $L^2(\Omega)$ denote the space of L^2 integrable vector functions on Ω with the standard L^2 inner product, and $H^1(\Omega)$ denote the Sobolev space so that the first derivative of L^2 integrable vector functions on Ω is also contained in $L^2(\Omega)$.*

Then

$\forall \mathbf{w} \in L^2(\Omega), \exists \nabla\phi \in L^2(\Omega)$ (*with* $\phi \in H^1(\Omega)$) *and* $\mathbf{u} \in L^2(\Omega)$ *such that*

$$\mathbf{w} = \nabla\phi + \mathbf{u} \quad (3.7a)$$

with

$$\nabla \cdot \mathbf{u} = 0 \quad (3.7b)$$

$$\nabla \times \nabla\phi = 0 \quad (3.7c)$$

Provided the boundary condition:

$$\mathbf{u} \text{ parallel to } \partial\Omega \text{ (i.e. } \mathbf{u} \cdot \mathbf{n} = 0\text{), on } \partial\Omega$$

Further as we shall see shortly the decomposition is orthogonal. This partly follows from the boundary condition provided in the theorem. The uniqueness then follows from the orthogonality condition.

Note that the boundary condition can actually be done in two ways. Let \mathbf{n} be the unit normal vector to $\partial\Omega$. First, intuitively, we can make \mathbf{u} parallel to $\partial\Omega$ by enforcing its normal component to vanish, i.e. make $\mathbf{n} \cdot \mathbf{u} = 0$ without specifying its tangential component; alternatively, although less direct we can make $\mathbf{n} \times \nabla\phi = 0$ so that the vector tangential component of \mathbf{u} is the same as the vector tangential component of \mathbf{w} , the normal component of \mathbf{u} is rather undetermined. In fact both ways will retain the orthogonality, existence and uniqueness of the decomposition. For the sake of simplicity, we only show the demonstration for the vanishing normal component case. However detailed proof for the other case can be found in Denaro's paper [21]. I point this out because the second case when $\mathbf{n} \times \nabla\phi = 0$ will lead to the potential vorticity formulation of projection method, which is another robust solver. We have not implemented this method in our numerical solver due to limited time. In addition, this implies that the Helmholtz decomposition only requires the specification of one boundary condition (normal or tangential) for \mathbf{w} . This is a controversial issue in projection methods, because the projected divergence free vector field \mathbf{u} (which is taken to be the velocity solution to the Navier Stokes equations) is not guaranteed to satisfy all the physical

boundary conditions that the Navier Stokes problem require [21].

With the boundary condition $\mathbf{n} \cdot \mathbf{u} = 0$ provided, we demonstrate the existence, uniqueness and orthogonality of the Helmholtz decomposition.

First the existence of decomposition:

By taking divergence on equation Eq. (4.1 a) we arrive at the following relations:

$$\nabla \cdot \mathbf{w} = \nabla^2 \phi, \quad \text{on } \Omega \quad (3.8a)$$

Then by taking the scalar normal component of \mathbf{w} along the boundary we obtain.

$$\begin{aligned} \mathbf{n} \cdot \mathbf{w} &= \mathbf{n} \cdot \nabla \phi \\ &= \frac{\partial \phi}{\partial \mathbf{n}}, \quad \text{on } \partial\Omega \text{ provided } \mathbf{u} \cdot \mathbf{n} = 0 \end{aligned} \quad (3.8b)$$

This is essentially a Poisson problem over variable ϕ with a Neumann boundary condition given.

This is a well studied mathematical problem and as proved by R. Courant and D. Hilbert in 1953, [9, 10, 21] this problem has a unique solution (up to the addition of a constant) provided $\int_{\Omega} f dV = \int_{\partial\Omega} g dS$. With the notation $f = \nabla^2 \phi$ and $g = \frac{\partial \phi}{\partial \mathbf{n}}$, this is rather immediate from the divergence theorem.

$$\begin{aligned} \int_{\Omega} f dV &= \int_{\Omega} \nabla^2 \phi dV \\ &= \int_{\Omega} \nabla \cdot (\nabla \phi) dV \\ &= \int_{\partial\Omega} \nabla \phi \cdot \mathbf{n} dS \\ &= \int_{\partial\Omega} g dS \end{aligned} \quad (3.8c)$$

We now demonstrate the orthogonality property of the decomposition.

First notations: followed by the decomposition, let V denote the subspace of $L^2(\Omega)$ that contains the curl free vector fields ($\nabla \phi$, with $\phi \in H^1(\Omega)$) and H be the subspace of $L^2(\Omega)$ that contains all the divergence free vector fields (\mathbf{u}).

Hence we can write H formally as:

$$H = \{\mathbf{u} \in L^2(\Omega) : \nabla \cdot \mathbf{u} = 0, \mathbf{u} \cdot \mathbf{n}|_{\partial\Omega} = 0\} \quad (3.9)$$

The orthogonality property is shown in the sense that the inner product between any elements in V and H is zero. Indeed orthogonality depends on the structure of the vector space we are working with. In our case, we have $L^2(\Omega)$, $\Omega \subset \mathbb{R}^3$ with the standard L^2 inner product defined as:

$$\langle \mathbf{w}, \mathbf{v} \rangle = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} dV$$

for any \mathbf{w} and $\mathbf{v} \in L^2(\Omega)$.

The demonstration of orthogonality is illustrated below
 $\forall \nabla\phi \in V$ (with $\phi \in H^1(\Omega)$) and $\mathbf{u} \in H$ and by the decomposition.

$$\int_{\Omega} \nabla\phi \cdot \mathbf{u} dV = \int_{\Omega} \nabla \cdot (\phi\mathbf{u}) - \int_{\Omega} \phi(\nabla \cdot \mathbf{u}) dV \quad (3.10a)$$

this is done by the vector calculus identity: Product rule of a scalar and vector field:

$$(\nabla \cdot (A\mathbf{B})) = \mathbf{B} \cdot \nabla A + A(\nabla \cdot \mathbf{B}), \text{ where } A = \phi, \mathbf{B} = \mathbf{u}$$

Because by decomposition, \mathbf{u} is divergence free, hence the second integral vanishes.

By applying Divergence Theorem to the remaining integral in the equation above

$$\begin{aligned} \int_{\Omega} \nabla\phi \cdot \mathbf{u} dV &= \int_{\partial\Omega} \phi\mathbf{u} \cdot \mathbf{n} dS \\ &= 0, \quad \text{by construction } \mathbf{u} \cdot \mathbf{n} = 0 \end{aligned} \quad (3.10b)$$

We now give a simple demonstration of uniqueness.

Suppose that the decomposition is not unique.

Then $\exists \nabla\phi, \nabla\phi' \in V$ (with $\phi, \phi' \in H^1(\Omega)$) and $\mathbf{u}, \mathbf{u}' \in H$, such that they both constitute valid Helmholtz decompositions (Note that the prime sign ' is just used to distinguish between vector fields). Hence

$$\mathbf{w} = \nabla\phi + \mathbf{u},$$

$$\mathbf{w} = \nabla\phi' + \mathbf{u}'$$

with $\mathbf{n} \cdot \mathbf{u} = \mathbf{n} \cdot \mathbf{u}' = 0$

Taking their difference we obtain:

$$(\nabla\phi - \nabla\phi') + (\mathbf{u} + \mathbf{u}') = 0 \quad (3.11)$$

Then taking an inner product with $(\nabla\phi - \nabla\phi')$

(exactly the same result holds if we do an inner product with $(\mathbf{u} - \mathbf{u}')$)

We then have

$$\begin{aligned} & \int_{\Omega} (\nabla\phi - \nabla\phi' + \mathbf{u} + \mathbf{u}') \cdot (\nabla\phi - \nabla\phi') dV = 0 \\ &= \int_{\Omega} (\nabla\phi - \nabla\phi') \cdot (\nabla\phi - \nabla\phi') + (\mathbf{u} - \mathbf{u}') \cdot (\nabla\phi - \nabla\phi') dV \quad (3.12a) \\ &= \int_{\Omega} \|\nabla\phi - \nabla\phi'\|_2^2 + (\mathbf{u} \cdot \nabla\phi - \mathbf{u}' \cdot \nabla\phi - \mathbf{u} \cdot \nabla\phi' + \mathbf{u}' \cdot \nabla\phi') dV \end{aligned}$$

where $\|\cdot\|_2$ denotes the Euclidean L_2 norm

with the help of orthogonality: ($\langle \nabla\phi \cdot \mathbf{u} \rangle = 0$ and $\langle \nabla\phi' \cdot \mathbf{u}' \rangle = 0$), the above equation simplifies to

$$\int_{\Omega} \|\nabla\phi - \nabla\phi'\|_2^2 - (\mathbf{u}' \cdot \nabla\phi + \mathbf{u} \cdot \nabla\phi') dV = 0 \quad (3.12b)$$

with the vanishing normal boundary condition: $\mathbf{n} \cdot \mathbf{u} = 0$ on $\partial\Omega$

$$\begin{aligned} \int_{\Omega} \mathbf{u}' \cdot \nabla\phi dV &= \int_{\Omega} \nabla \cdot (\phi\mathbf{u}') - \phi \nabla \cdot \mathbf{u}' dV \\ &= \int_{\partial\Omega} \phi \mathbf{u}' \cdot \mathbf{n} dV, \quad \text{by apply divergence theorem} \\ &= 0 \end{aligned} \quad (3.12c)$$

$\int_{\Omega} \mathbf{u} \cdot \nabla\phi' dV = 0$ is done using exactly the same method.

Hence this implies

$$\begin{aligned} &= \int_{\Omega} \|\nabla\phi - \nabla\phi'\|_2^2 dV \\ &= 0 \end{aligned} \quad (3.12d)$$

By the property of L^2 norm, this is true if and only if $\nabla\phi = \nabla\phi'$.

Therefore this implies $\mathbf{u}' = \mathbf{u}$. The decomposition is unique.

Hence we have now demonstrated the uniqueness, existence and orthogonality property of the Helmholtz Hodge Decomposition Theorem.

3.4 Application of Helmholtz Hodge Decomposition Theorem to numerical solution of Navier Stokes Equations

In this subsection, we will talk about the use of Helmholtz decomposition theorem to the construction of Projection methods in solving incompressible Navier Stokes equations. We are focused on the primitive Projection method first proposed by Chorin and independently by Temam [6, 7, 29]. Its advantages and disadvantages are also discussed.

3.4.1 Chorin's Original projection method

Recall the momentum part of the non-dimensionalised Navier Stokes equations

$$\mathbf{u}_t + \nabla p = -\mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{Re} \Delta \mathbf{u} \quad (3.13)$$

Chorin realised that the left hand side of the momentum equation is actually in the form of the Helmholtz Hodge Decomposition [6, 7, 9]

$$\mathbf{w} = \mathbf{u}_t + \nabla p \quad (3.14)$$

and as to obtain an unique Helmholtz Hodge decomposition we also require that \mathbf{u}_t parallel to $\partial\Omega$ on the boundary $\partial\Omega$ and \mathbf{u}_t is orthogonal with ∇p . It is clear that this imposes non-physical boundary conditions as in many cases, the normal component of velocity ($\mathbf{u}_t \cdot \mathbf{n}$) is not zero! This is a controversial issue and some researchers at first doubted the validity of such constructions [6, 23, 26]. However as proved later by people including Chorin himself that the method is indeed convergent and accurate to at least first order [6, 8, 23, 25, 26].

For now, let's assume \mathbf{u}_t is the divergence free component followed by the Helmholtz decomposition.

Hence we note \mathbf{u}_t belongs to the set H (Equation 4.9)

The above discussion naturally lead us to define a projection operator \mathbb{P} which eliminates the gradient of pressure term and left the projected vector \mathbf{w} to be divergence free. Hence the projection operator maps the vector \mathbf{w} into the divergence free vector space H .

Definition 1. Let H (as defined in Equation. 4.3) and V be two orthogonal subspaces of $L^2(\Omega)$ such that $L^2(\Omega) = H \oplus V$.

Then the projection operator \mathbb{P} is a linear map defined as follows:

$$\forall \mathbf{w} \in L^2(\Omega)$$

$$\mathbb{P}(\mathbf{w}) : L^2(\Omega) \rightarrow H$$

with the boundary condition required by Helmholtz Decomposition:

$$\mathbf{w}_1 = \mathbb{P}(\mathbf{w}) \text{ parallel to } \partial\Omega$$

We can also define \mathbb{P} formally as:

$$\mathbb{P} = I - \nabla(\nabla \cdot \nabla)^{-1}\nabla \cdot$$

Thus V can be written as:

$$V = \{\mathbf{w}_2 \in L^2(\Omega) : \mathbb{P}(\mathbf{u}) = 0\}$$

(recall from the Helmholtz theorem in previous section, $\mathbf{w}_1 = \mathbf{u}$ and $\mathbf{w}_2 = \nabla\phi$, $\phi \in H^1(\Omega)$)

By construction, V and H are the Kernel and Range of \mathbb{P} respectively .

Below are some of the basic properties \mathbb{P} has:

$$\forall \mathbf{w}_1 \in H, \mathbf{w}_2 \in V; \forall \mathbf{w} \text{ and } \mathbf{z} \in L^2(\Omega)$$

\mathbb{P} is the identity operator on H :

$$\begin{aligned} \mathbb{P}(\mathbf{w}_1) &= \mathbf{w}_1 - \nabla(\nabla \cdot \nabla)^{-1}\nabla \cdot \mathbf{w}_1 \\ &= \mathbf{w}_1 \end{aligned} \tag{3.15a}$$

since $\nabla \cdot \mathbf{w}_1 = 0$.

In addition V is the null set of \mathbb{P}

$$\begin{aligned} \mathbb{P}(\mathbf{w}_2) &= \nabla\phi - \nabla(\nabla \cdot \nabla)^{-1}\nabla \cdot (\nabla\phi) \\ &= \nabla\phi - \nabla(\nabla \cdot \nabla)^{-1}(\nabla \cdot \nabla)\phi \\ &= \nabla\phi - \nabla\phi \\ &= 0 \end{aligned} \tag{3.15b}$$

\mathbb{P} is idempotent because $\mathbb{P}(\mathbb{P}(\mathbf{w})) = \mathbb{P}(\mathbf{w})$

\mathbb{P} is linear since $\mathbb{P}(\mathbf{w} + \mathbf{z}) = \mathbb{P}(\mathbf{w}) + \mathbb{P}(\mathbf{z})$

\mathbb{P} is orthogonal projection by construction

\mathbb{P} is also self adjoint

3.4. APPLICATION OF HELMHOLTZ HODGE DECOMPOSITION THEOREM TO NUMERICAL METHODS

For the Navier Stokes equations $\mathbf{u}_t \in H$ and $\nabla p \in V$ because \mathbf{u}_t is divergence free.

Hence if we apply \mathbb{P} to the momentum equation we obtain:

$$\mathbb{P}(\mathbf{u}_t + \nabla p) = \mathbf{u}_t = \mathbb{P}(-(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{1}{Re}\Delta\mathbf{u}) \quad (3.16)$$

Thus we have eliminated the Pressure term and decoupled the momentum equation. This is the fundamental idea of the Projection method. It is not only of theoretical interest to the analysis of Navier Stokes equations (e.g. see [12, 31]) but also shedding light to the practical use in computing numerical solutions [6, 7, 29].

In this chapter we are mainly interested in constructing numerical solution using projection method. The task is at each iteration: given a divergence free velocity field \mathbf{u} which satisfies the momentum equation at time n and the correct boundary condition required by the problem, we perform the projection to update the velocity to time $n+1$ and still satisfies all the constraints.

the Numerical discretisation is:

For $n = 0, 1, \dots, N$, where $N \in \mathbb{N}$, the time step is: $\Delta t = \frac{T_f - T_0}{N}$ where T_f, T_0 denotes the end time and start time respectively.

$$\mathbf{u}_t^n = \mathbb{P}((-\mathbf{u}^n \cdot \nabla)\mathbf{u}^n + \frac{1}{Re}\Delta\mathbf{u}^n) \quad (3.17)$$

where \mathbf{u}^n denote the discretised velocity field at time $n\Delta t$.

We now introduce the spatial discrete version of the projection operator

$$\mathbb{P} = I - \nabla_h(\nabla_h \cdot \nabla_h)^{-1}\nabla_h \cdot \quad (3.18)$$

where I is the identity matrix and $\nabla_h \cdot$ and ∇_h are the discrete approximation to divergence and gradient operators.

As first developed by Chorin, this is called the *Exact Projection method* [1, 2, 7] because this is the direct discretisation of the projection operator (see *Definition 1*). However this often introduces numerical instabilities especially in going to the limit of zero Mach number in reacting flows [1, 2, 19, 22], and hence it was replaced by the so called *Approximate Projection methods* from the start of 1990s [1, 2, 6]. We will talk about the details in subsection *Variations in Projection Methods*

It is important to note that neither the diffusive nor the convective terms in Equation 4.10 belong to V or H . $-\mathbf{u} \cdot \nabla \mathbf{u}$ is neither curl or divergence free and

while $\frac{1}{Re}\Delta\mathbf{u}$ is divergence free but it may not be parallel to the boundary. Hence we cannot simply expand the right hand side of Equation 4.10.

We can get around this problem by introducing an auxiliary (or intermediate) vector field \mathbf{u}^* in between the time step n and $n+1$ such that its derivative equals to the terms inside the projection operator in the right hand side of Equation 4.10:

$$\begin{aligned}\mathbf{u}_t^* &= \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} \\ &= (-\mathbf{u}^n \cdot \nabla)\mathbf{u}^n + \frac{1}{Re}\Delta\mathbf{u}^*\end{aligned}\tag{3.19}$$

with backward Euler finite difference in time.

Note that the auxiliary field \mathbf{u}^* does not satisfy the continuity constraint and it is simply used to advance to \mathbf{u}^{n+1} which analytical should satisfy the continuity equation. \mathbf{u}^* is discarded after each iteration.

The dilemma here is: we start with a vector field (\mathbf{u}^*) satisfies the correct boundary condition but is not divergence free. Then we update to the velocity field to the next time level by projecting \mathbf{u}^* to the divergence free space. However the updated velocity now generally don't satisfy the correct boundary condition anymore because of the condition that \mathbf{u} be parallel to $\partial\Omega$ on the boundary. For instance, in a 2D incompressible flow problem with homogeneous Dirichlet boundary condition: $\mathbf{u}|_{\partial\Omega} = 0$, the numerical velocities \mathbf{u}^n might have non-zero tangential components in either of the two cases required by unique HHD. Therefore the projection method or at least the primitive version is inherently less accurate than the more cumbersome coupled iterative solvers. As discussed in the subsequent sections, we can fix this problem through imposing appropriate boundary constraints and better pressure correction [6].

Assume $\mathbf{u}_t^* \in L^2(\Omega)$ we can therefore project it onto the space of divergence free vector fields (V) and decompose by the Helmholtz Hodge Decomposition

3.4. APPLICATION OF HELMHOLTZ HODGE DECOMPOSITION THEOREM TO NUM

theorem (HHD):

$\mathbb{P}(\mathbf{u}_t^*) = \mathbf{u}_t$, where \mathbf{u}_t is divergence free, so

$$\begin{aligned}\mathbf{u}_t^* &= \mathbf{u}_t + \nabla\phi, \text{ The smooth representation of the decomposition} \\ \rightarrow \mathbf{u}_t - \mathbf{u}_t^* &= \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} - \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} \\ &= -\nabla\phi\end{aligned}\tag{3.20}$$

finally

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \Delta t \nabla\phi$$

where $\nabla\phi \in V$ is a curl free vector field resulting from HHD.

In Chorin's original work ϕ is simply the approximation to the pressure at time $n+1$ [7]. Usually p^n is a good guess as it satisfies the boundary conditions specified. We will soon see that this will inherently give first order accuracy to pressure along the boundary $\partial\Omega$. However, at the moment, we are just trying to illustrate the basic algorithm for the primitive projection method.

Replacing ϕ by p^{n+1} and using the spatial discrete projection operator the decomposition in Eq. () becomes:

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \Delta t \nabla_h p^{n+1}\tag{3.21}$$

In Exact Projection method, $\nabla_h \cdot \mathbf{u}^{n+1} = 0$ when the exact approximation of pressure at time $n+1$ used. However because we don't have access to the exact p^{n+1} (rather p^n), hence as argued by Chorin [7] to ensure convergence of the primitive variables (at time $n+1$) the above decomposition is best done by using multi-step iterative methods.

Let m denote the number of iterations, we replace Equation 4.15 by a multi-steps iterative scheme:

$$\mathbf{u}^{n+1,m+1} = \mathbf{u}^* - \Delta t \nabla_h f^m(p),, \text{ on } \Omega \text{ only}\tag{3.22a}$$

$$p^{n+1,m+1} = p^{n+1,m} - \lambda \nabla_h \cdot \mathbf{u}^{n+1,m+1},, \text{ on } \Omega \text{ and } \partial\Omega\tag{3.22b}$$

where f^m is a function of $p^{n+1,m+1}$ and $p^{n+1,m}$ which converges to p^{n+1} as $|p^{n+1,m+1} - p^{n+1,m}| \rightarrow 0$; λ is a constant to be determined. Hence this ensures Equation 4.16 a converges to Equation 4.15 over successive iterations

To start the iteration we define $p^{n+1,m=1} = p^n$

By substituting Equation 4.16 into Equation 4.16 b we obtain

$$p^{n+1,m+1} - p^{n+1,m} = -\lambda \nabla_h \cdot \mathbf{u}^* + \Delta t \nabla_h \cdot (\nabla_h f^m(p))\tag{3.22c}$$

Note this condition only holds in interior points.

Chorin argues that if $|p^{n+1,m+1} - p^{n+1,m}| \rightarrow 0$ then this is equivalent of solving the Poisson equation:

$$\nabla_h^2 p^{n+1} = \nabla_h \cdot (\nabla_h p^{n+1}) = \frac{1}{\Delta t} \nabla_h \cdot \mathbf{u}^* \quad (3.23)$$

where ∇_h^2 is the discrete Laplace operator.

As Equation 4.16 a is simply not defined along the boundary, appropriate boundary conditions must be chosen for p . Note by taking the scalar normal component of Equation 4.15

$$\begin{aligned} \mathbf{n} \cdot \mathbf{u}^* &= \mathbf{n} \cdot \mathbf{u}^{n+1} + \mathbf{n} \cdot \nabla_h \phi^{n+1}, \quad \mathbf{u}^{n+1} \in H \\ &= \mathbf{n} \cdot \nabla_h \phi^{n+1} \end{aligned}$$

we conclude that the pressure approximation satisfies a Neumann boundary condition:

$$\mathbf{n} \cdot \nabla p^n = \frac{\partial p}{\partial n} |_{\partial\Omega} = \frac{1}{\Delta t} \mathbf{n} \cdot \mathbf{u}^* |_{\partial\Omega}$$

This non - physical boundary has caused some controversial discussions about the accuracy of projection method and questioned about the curl free field approximation to the true pressure value [25, 26]. This indeed affect the accuracy of projection especially along the boundary, but at least the original projection method as introduced by Chorin is at most first order accurate in time for both velocity and pressure [6, 25, 26]. Higher order schemes were then devised in the later half of the century, These will be added in the subsection *variations in projection methods*.

Hence we need to choose the function $f^m P$ and λ so that Equation 4.16 c is a rapidly converging iteration to solve the Poisson equation Equation 4.17 [7]. They can be specified after the spatial discretisation is done. However it is not our purpose here to present Chorin's full original method. (details can be found in his original paper [7]

A standard version of Chorin's Projection method can be summarised into the following steps:

Step 1: Calculate the intermediate velocity field

$$\begin{aligned} \mathbf{u}^* &= \mathbf{u}^n + \Delta t (-\mathbf{u}^n \cdot \nabla) \mathbf{u}^n + \frac{1}{Re} \nabla_h^2 \mathbf{u}^n \\ \mathbf{u}^* |_{\partial\Omega} &= \mathbf{u}((n+1)\Delta t) |_{\partial\Omega} \end{aligned}$$

3.4. APPLICATION OF HELMHOLTZ HODGE DECOMPOSITION THEOREM TO NUMERICAL METHODS

Step 2: Perform the Projection

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \Delta t \nabla_h p^{n+1}$$

Step 3: Update pressure and velocity

$$\begin{aligned}\nabla^2 p^{n+1} &= \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad \frac{\partial p}{\partial n}|_{\partial\Omega} = \frac{1}{\Delta t} \mathbf{n} \cdot \mathbf{u}^*|_{\partial\Omega} \\ \mathbf{u}^{n+1} &= \mathbf{u}^* - \Delta t \nabla_h p^{n+1}\end{aligned}$$

where the Laplacian operator ∇_h^2 is approximated by $\nabla_h \cdot (\nabla_h)$

Chorin's exact Projection method although been widely used only showed first order convergence in time [6, 7, 25, 26].

Chorin has analysed the accuracy of his scheme only for periodic boundaries [8] and E. Liu, Jie Shen and Rannacher [20, 25, 26] and many other authors have extend this analysis to a general case.

The accuracy in velocity can be readily improved by using higher order numerical methods (e.g. Crank Nicholson method \dots), the pressure on the other hand is not so straightforward. It has even been suggested by authors like Perot that the Projection method is inherently first order accurate for pressure [23]. Later we will see in the next section that we can lift up the accuracy in pressure by considering more advanced variations.

3.5 Variations in Projection methods

Since the birth of projection method in the 1960s, numerous effort has been made by researchers to improve the accuracy, especially for pressure. In this section we will consider the popular methods described in literature and then analyse the ingredients that are necessary to obtain fully second order schemes in all primitive variables.

3.5.1 Approximate Projection method

First, we encounter the idea of “Approximate” projection method as opposed to the “Exact” method in Chorin’s original method.

The exact projection method means the final computed velocity must satisfy a discrete divergence constraint by applying the discrete orthogonal projection operator. There are however serious drawbacks of the Exact Projection method. The major problem comes from the discrete Laplace operator. In Chorin’s original work, the Laplace operator ∇_h^2 is approximated by $\nabla_h \cdot (\nabla_h)$ which is the composition of discrete divergence and gradient operators [2, 7]. Let’s denote this discrete Laplacian as L_E . It has some nice properties such as skew adjoint ($(\nabla_h \cdot = -\nabla_h^T)$) and idempotent ($L_E^2 = L_E$) [1, 2]. A vertex centred grid was used in Chorin’s original paper where both velocities and pressures are specified at grid vertices [1, 2, 7]. Note this is not the Non - staggered grid where velocities and pressure values are placed at the same location. With a centred finite difference discretisation applied to both D and G , an expanded 5 point stencil with $2h$ spacing is formed (for simplicity, we let $h = \Delta x = \Delta y$).

We illustrate L_E below in a 2D context with a squared vertex centred grid. Let $\mathbf{u}_{i,j} = (u_{i,j}, v_{i,j})$ denote the discrete velocities at interior node location (i,j) and at an arbitrary time step. For spatial discretisation, i denotes the column (hence horizontal direction) and j denote the row (vertical direction) (**Need a graph of the grid!**).

(2nd order accurate) centred finite difference applied to

Divergence (of velocities)

$$\nabla_h \cdot (\mathbf{u}_{i,j}) = \frac{1}{2\Delta x}(u_{i+1,j} - u_{i-1,j}) + \frac{1}{2\Delta y}(v_{i,j+1} - v_{i,j-1}) \quad (3.24a)$$

$$\partial_x(\phi_{i,j}) = \frac{1}{2\Delta x}(\phi_{i+1,j} - \phi_{i-1,j}), \quad \text{Horizontal direction} \quad (3.24b)$$

$$\partial_y(\phi_{i,j}) = \frac{1}{2\Delta y}(\phi_{i,j+1} - \phi_{i,j-1}), \quad \text{Vertical direction} \quad (3.24c)$$

As a composite function $L_E = \nabla_h \cdot (\nabla_h \phi)$ can be therefore expressed by substituting Equatino 4.18 b, c into Equation 4.18 a.

$$L_E = \frac{1}{4h^2}(\phi_{i-2,j} + \phi_{i+2,j} - 4\phi_{i,j} + \phi_{i,j-2} + \phi_{i,j+2}) \quad (3.25)$$

This is different to the more standard 5 - point stencil to Laplace which is formed by directly approximate to the second derivatives.

$$L_A = \frac{1}{h^2}(\phi_{i-1,j} + \phi_{i+1,j} - 4\phi_{i,j} + \phi_{i,j-1} + \phi_{i,j+1}) \quad (3.26)$$

where A stands for "Approximation" and we will soon what this notation means.

There are a lot of shortcomings for the wide 5 point stencils (L_E Equation 4.19) including more cells need to be implemented in order to calculate derivatives along the boundary and also introducing weak instabilities as a result of local grid decoupling [1, 2, 6, 17, 22].

This wide 5 point stencil has no coupling (dependence) between adjacent nodes. This is referred to local grid decoupling and was analysed by many researchers in the late 1990s [1, 2, 17, 22].

As a result of the decoupling, the spatial grid can now be separated into 4 different non-interactive sub-grids [17, 22]. These sub-grids are locally isolated in a way that the value of $L_{E(i,j)}$ depends only on the grid points inside. The sub-grids are independent of each other only except along the boundary.

This local grid decoupling raises many issues. First it makes the implementation of iterative solvers like multigrid method difficult [1, 2, 17]. Alterations to the stencil structure is needed. Second the null space of L_E contains oscillatory mode which cannot be easily removed by the projection operator when applied to velocities [22] (especially for flow resulting a sharp velocity gradient). Hence in situations like low Mach flow, marked oscillations observed [19].

There have been many methods proposed to get around this problem, For instance, by Bell, Colella and Glaz introduced a finite element discretisation on staggered grids [5] (pressure at centre and velocities at nodes) of L_E resulting a more compact stencil. However local grid decoupling is still preserved [1, 2].

Other methods trying to alter the wide stencil to accommodate (large and asymmetrical) so that multi-grid operator can be applied [17].

Another very popular method originally introduced Harlow and Welch [16] was to use a staggered grid structure. Instead of node centred or cell centred, MAC (or staggered) grid is more like a combination of the two. The horizontal velocity (u) is stored on the vertical edges and vertical velocities (b) is stored on the horizontal edges while the pressure node is stored at the cell centres. As a result, the more compact stencil L_A can be used and decoupling is avoided. However, because the horizontal and vertical velocities are now stored in different locations and hence makes calculations of the non-linear convections cumbersome [2, 17]. While transferring of velocity nodes to the cell centre is possible but this would make it impossible to find a discrete divergence $\nabla_h \cdot$ such that $\nabla_h \cdot u = 0$.

Approximation methods proposed by Almgren et al 1996 [2] provided a solution to this problem by relaxing the divergence constraint from a theoretical zero value to the truncation error of the method.

Hence for a standard second order accurate scheme we have

$$\nabla_h \cdot \mathbf{u} = O(h^2)$$

A direct approximation to the Laplacian (Δ) is therefore used and now $\nabla_h^2 \approx \nabla_h \cdot (\nabla_h)$. However the projection operator (below) is now not idempotent.

$$\mathbb{P}_A = I - G(L_A^{-1})D \quad (3.27)$$

This method avoids the use of the wide L_E stencil while also supports the cell centred scheme which will be used in computing the non-linear convection terms. This method is shown to be very effective for more complicated spatial geometry and especially on adaptive mesh [17]

The procedure for an approximate projection method is essentially the same as the exact method except the 5 point standard stencil L_A is used.

In practice, modern projection methods combines the staggered grid with the approximation method to maximise performance [6].

In the following section, we consider recent variations in Projection methods with the aim of obtaining a fully second order accuracy in both space and time

for all primitive variables. The approximation projection operator (\mathbb{P}_A) is used.

3.5.2 Second order accurate schemes

At the time of construction, Chorin's Projection method and especially its efficiency was still considered to be a great progress in computational fluid dynamics. However there are still many drawbacks associated with it. In particular, as noted before there are a lot of controversial discussion about the choice of boundary conditions. For instance the intermediate velocity (\mathbf{u}^*) and the projected velocity; the choice of specification of normal and tangential components of velocity boundaries. Among those issues, the most concerned and also the most critical issues is how should the pressure term to be recovered from the projection. This will soon to be shown to have determining impact on the overall accuracy of the method.

As discussed in the previous section, Chorin's projection method was only shown to be first order accurate. Certainly Chorin's work has left space for improvements, however it was not until more than 20 years later when academics were able to obtain seconder order or semi-seconder order schemes (e.g. [5, 18]). The generalisation to seconder order accuracy in time for velocity variables were not difficult to achieve. However it is the pressure and especially along the boundary layer which causes problem.

Let us consider a general setup of the numerical scheme which most modern Projection methods use.

Consider a second order time centred differencing scheme. Thus we are solving the primitive variables at time $n + 1/2$ step. This was proposed by a number of authors in 1980s including Goda [13], Bell [5], Kim and Moin [18] and Van Kan [32]

$$\mathbf{u}_t^{n+1/2} = \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1/2} = -[(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+1/2} + \frac{1}{R} \Delta \mathbf{u}^{n+1/2} \quad (3.28a)$$

second order Crank-Nicholson scheme is used to discretise the Diffusion term. For simplicity, the convection term is not being fully discretised for now.

We therefore arrived at the numerical scheme of:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1/2} = -[(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+1/2} + \frac{1}{2R} \Delta (\mathbf{u}^{n+1} + \mathbf{u}^n) \quad (3.28b)$$

with the divergence constraint

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (3.28c)$$

and boundary condition

$$\mathbf{u}^{n+1} = \mathbf{u}((n+1)\Delta t)|_{\partial\Omega} \quad (3.28d)$$

the subscript “h” for discrete operators is dropped for simplicity.

It is worth to note that second order schemes other than centred finite difference and Crank Nicholson for implicit diffusion term would yield the same results in terms of accuracy (For instance, Pyo and Shen have used second-order backward difference formula (BDF2) for time derivative approximations) [24]. We present the above algorithm because it has been implemented in our numerical solver.

Numerical steps of this modified Projection method:

Step 1:

Solve for intermediate velocity field \mathbf{u}^*

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \nabla q = -[(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+1/2} + \frac{1}{2Re} \Delta(\mathbf{u}^* + \mathbf{u}^n) \quad (3.29a)$$

This is identical to the original scheme except that we are using a centred time differencing and a (curl free) scalar potential q to approximate the pressure $p^{n+1/2}$. Boundary condition for \mathbf{u}^* also need to be specified. We will consider varies choices later

Step 2:

Perform the Projection (simply decompose the intermediate velocity field according to the Helmholtz Decomposition Theorem. Identical to the original method.)

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \Delta t \nabla \phi^{n+1} \quad (3.30a)$$

(Note ϕ is not an approximation to $p^{n+1/2}$, it is just a term resulting from the Helmholtz decomposition)

And the divergence constraint (uses Approximation method)

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \text{ only up to the truncation error: } \mathcal{O}(\Delta t^2) \quad (3.30b)$$

Calculate ϕ^{n+1} by solving a Poisson equation

$$\nabla^2 \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \quad (3.30c)$$

subject to the Neumann boundary condition

$$\mathbf{n} \cdot \nabla \phi|_{\partial\Omega} = \mathbf{n} \cdot (\mathbf{u}^* - \mathbf{u}^{n+1})|_{\partial\Omega} \quad (3.30d)$$

Step 3:

Pressure velocity correction (the most critical part of the Projection method)

$$p^{n+1/2} = q + L(\phi^{n+1}) \quad (3.31a)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \phi^{n+1} \quad (3.31b)$$

where L is a function of ϕ such that the pressure can be correctly updated. We will see soon how this would impact the accuracy of pressure.

This is often referred as incremental pressure projection method [6] as the projection step and step 3 works to compute an incremental pressure correction each time. This is opposed to the original projection method proposed by Chorin where the pressure is fully recovered from the Poisson equation and no correction is being made thereafter.

In addition, as mentioned before, the non-physical boundary condition: $\mathbf{n} \cdot \mathbf{u}^{n+1} = 0$ in Chorin's original method is a direct requirement from Helmholtz decomposition. In practice, we really want the projected divergence free vector field \mathbf{u}^{n+1} to satisfy the correct boundary condition required by the problem so that it satisfies the Navier Stokes equations up to the boundary. Hence we really need to re-define the projection operator. We relax the the boundary condition in the Helmholtz decomposition theorem, so that the divergence free vector space is re-defined as

$$H' = \{\mathbf{u} \in L^2(\Omega) : \nabla \cdot \mathbf{u} = 0, \mathbf{u} \cdot \mathbf{n}|_{\partial\Omega} = \mathbf{u}_b\}$$

where now the boundary condition for the divergence free vector field is not zero. It is usually taken be the exact boundary condition of the true velocity solution of the Navier Stokes equations.

Note now, the new projection operator (\mathbb{P}) defined based on H' may not be orthogonal any more. Hence the resulting decomposition may not be unique either. The question of whether orthogonality is necessary still remains open [21]. Denaro has proposed an extra Helmholtz decomposition in the projection step to

ensure orthogonality. However this corresponds to 2 successive Poisson equations to be solve, which is quite expensive. Therefore we do not consider it here for the sake of efficiency [21]. The question of whether orthogonality is important still remains open. We show in chapter 5 that this should not be an issue in terms of accuracy at least in simple domains like periodic channel.

Hence, from now on we use the modified projection operator \mathbb{P} .

To achieve optimal convergence rate, there are 3 things that need to be considered: the pressure approximation q , the boundary condition for the intermediate velocity and the pressure correction function L . Brown has argued that the coupling between these 3 issues must be considered to obtain high order schemes [6]. To improve the accuracy in pressure, Brown has proposed a new pressure update formula shown below. [6].

Substitute Equation 4.24 a into Equation 4.23 a to eliminate \mathbf{u}^* and compare to the centred differencing scheme Equation 4.22 b we arrived at the pressure update formula:

$$p^{n+1/2} = q + \phi^{n+1} - \frac{\Delta t}{2Re} \nabla^2 \phi^{n+1} \quad (3.32a)$$

Hence

$$L(\phi^{n+1}) = \phi^{n+1} - \frac{\Delta t}{2Re} \nabla^2 \phi^{n+1} \quad (3.32b)$$

the last term of the function L is critical to ensuring a second order accurate scheme to pressure and many previous methods failed to be second order accurate because they did not involve this correction term.

The role of the intermediate velocity \mathbf{u}^* still remains undetermined. In the original Projection method proposed by Chorin, it is simply served to approximate the fluid velocity at an intermediate time (between n and $n+1$). However the question is not so simple in the incremental pressure projection method. This question must be answered by taking into account the role of q . If q is a good approximation to $p^{n+1/2}$ (to the truncation error of the method), then \mathbf{u}^* should not deviate from the fluid velocity very much and vice versa.

Boundary condition is always critical. Notice that the intermediate velocity is closely related to the divergence free velocity and the gradient term (ϕ) by the decomposition given in Equation 4.24 a. Hence the boundary condition of \mathbf{u}^* must be consistent with that of \mathbf{u}^{n+1} and ϕ^{n+1} even though ϕ^{n+1} is not known

yet! Hence an appropriate approximation to it must be considered and this is a question that have puzzled many researchers [6]. In fact all these issues are related to q and the degree of approximation to the true pressure field.

Naturally people would think that the pressure at the previous time would be a good approximation. Hence let's choose $q = p^{n-1/2}$. This type of incremental pressure projection method was first proposed by *Bell, Colella and Glaz* [5]. Hence the intermediate velocity is closer to the true fluid velocity and thus by ensuring the same boundary condition we obtained:

$$(\mathbf{u}^* - \mathbf{u}((n+1)\Delta t))|_{\partial\Omega} = 0 \quad (3.33a)$$

A convenient choice is simply let $(\mathbf{u}^* = \mathbf{u}((n+1)\Delta t))|_{\partial\Omega}$. Because then we have a zero Neumann boundary condition for ϕ^{n+1} and it is simple to implement.

$$\mathbf{n} \cdot \nabla \phi^{n+1}|_{\partial\Omega} = 0 \quad (3.33b)$$

It is worth to point out that the original scheme proposed by Bell uses a rather different pressure update formula [5]:

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1} \quad (3.34)$$

which is inherently first order accurate [6].

Let's denote this as “**Alg 1**” which simply means algorithm 1.

This loss of accuracy in the pressure, which typically manifests itself as a numerical boundary layer, is well known and has been analyzed rigorously by Temam [30], E and Liu [20], Shen [27], and many others.

As noted by Brown that this problem can be solved if the modified pressure update formula Equation 4.26 a is used instead and this would recover a full second order scheme for both pressure and velocity up to boundary [6]. We will see this in chapter 5 using a normal mode analysis. Therefore let use denote such method as “**Alg 2**”. Note that “Alg 2” is identical to “Alg 1” except the different pressure update formula used.

There is another way of looking at the problem. Because of the limitation of accuracy of pressure update, what if we don't simply don't update pressure? Wouldn't it be nice if our calculations doesn't even involve pressure (and its gradient) at all? This might sound ridiculous at first because velocity and pressure are

strongly coupled by the divergence constraint. However this is not theoretically impossible. In fact it is very tempting to do it because we then would not have an accumulation of error of pressure in the numerical calculations. In 1985 Kim and Moin has proposed such a method with $q = 0$ in their well - cited paper [18]. This is referred to *The Pressure free Projection method*.

We can still choose the normal boundary component of \mathbf{u}^* to be equal to that of \mathbf{u}^{n+1} which leads to zero normal gradient for ϕ^{n+1} , the tangential component the intermediate velocity field needs more attention. Recall from projection step, the exact tangential boundary should be:

$$\tau \cdot \mathbf{u}^* = \tau \cdot (\mathbf{u}^{n+1} + \Delta t \nabla \phi^{n+1}) \quad (3.35)$$

In Alg 1 and 2 we have simply made $\tau \cdot \nabla \phi^{n+1} = 0$ too. We can not do this here because now the intermediate velocity field is not a second order approximation to \mathbf{u}^{n+1} (we shall see this through normal mode analysis and numerical tests) and so choosing the same boundary condition between \mathbf{u}^* and \mathbf{u}^{n+1} is not a wise choice. Because when solving for \mathbf{u}^* , we don't have access to ϕ^{n+1} yet, hence Kim has argued that using a lagged value: ϕ^n should do the job. We will see this provides second order accuracy to pressure in chapter 5.

The process of this pressure free projection method can be summarised as follows: Step 1:

Solve for intermediate velocity field \mathbf{u}^*

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -[(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+1/2} + \frac{1}{2R} \Delta(\mathbf{u}^* + \mathbf{u}^n) \quad (3.36a)$$

This is identical to the incremental pressure projection method Equation 4.23 a except that $q = 0$

$$\tau \cdot (\mathbf{u}^* - \nabla \phi^{n+1})|_{\partial\Omega} = \tau \cdot \mathbf{u}((n+1)\Delta t)|_{\partial\Omega} \quad (3.36b)$$

ϕ^n is used to approximate ϕ^{n+1} along the boundary

Step 2:

Solving for the gradient potential ϕ^{n+1} and update velocity

ϕ^{n+1} can be solved by the Poisson equation below with a zero Neumann boundary condition

$$\nabla^2 \phi^{n+1} = \frac{1}{\Delta t} \mathbf{u}^*, \quad \frac{\partial \phi^{n+1}}{\partial n}|_{\partial\Omega} = 0 \quad (3.37a)$$

velocity can therefore be updated to

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \phi^{n+1}, \quad \mathbf{u}^{n+1}|_{\partial\Omega} = \mathbf{u}((n+1)\Delta t)|_{\partial\Omega} \quad (3.37b)$$

The pressure update is ignored, however we can still use Equation 4.26 a to do it (except $q = 0$) if we are interested.

For convenience, we denote this as “**Alg 3**”.

3.5.3 Gauge method

To date, the projection methods we have considered are simply second order variations of Chorin’s original method. There are however also other type of “Projection methods” follows similar idea but of different structures. Gauge method is a popular one.

Gauge method or “Impulse” or “Magnetisation” methods was first introduced by Oseledets and then popularised by several researchers including E and Liu and Summers and Chorin [6, 33]. It provides an alternative option to decouple the velocity and pressure in the Navier Stokes equations by a change of variable. In this section we illustrates a standard second order Gauge method based on E and Liu as well as David.

The method starts by introducing a “Gauge” variable \mathbf{m} which is followed directly by the Helmholtz - Hodge decomposition:

$$\mathbf{m} = \mathbf{u} + \nabla\chi \quad (3.38)$$

where \mathbf{u} is the velocity field which satisfies the Navier Stokes equations and χ represents an auxiliary scalar potential obtained by projecting \mathbf{m} to the space of divergence free vector field. This is essentially the same as the standard Projection methods where \mathbf{m} replaces \mathbf{u}^* the intermediate velocity field. However \mathbf{m} and \mathbf{u}^* are not equal to each other! In fact we will soon see that the Gauge variable introduces advantages in accuracy.

To specify the value of χ we need to make sure the Navier Stokes equations are still satisfied after this change of variable. Substituting the expression for \mathbf{m} into the momentum equation we obtain:

(for simplicity the convective term is not transformed)

$$\partial_x \mathbf{m} - \partial_x(\nabla\chi) + (\nabla p + \nabla\chi) = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{1}{Re} \nabla^2 \mathbf{m}$$

Hence by ensuring the momentum equation is satisfied, we define the auxiliary field to satisfy the following relation with Pressure:

$$p = \partial_x \chi - \frac{1}{R} \nabla^2 \chi$$

Now we can discretise the scheme in time using second order schemes including centred finite differencing and Crank-Nicholson. We obtain:

$$\frac{\mathbf{m}^{n+1} - \mathbf{m}^n}{\Delta t} = -[(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+1/2} + \frac{1}{2 Re} \nabla^2 (\mathbf{m}^{n+1} + \mathbf{m}^n)$$

Hence the Gauge variable \mathbf{m} is not discarded but rather re-computed at each iteration. This is one of the major differences between the standard Projection methods and this Gauge method. Later through Normal analysis we will show the advantageous of such transformation. More specifically we show that by updating the Gauge variable the spurious mode is eliminated in all variables.

The first step is therefore solve \mathbf{m}^{n+1} with the knowledge of \mathbf{m}^n and \mathbf{u}^n and boundary condition:

$$\mathbf{n} \cdot \mathbf{m}^{n+1} |_{\partial\Omega} = \mathbf{n} \cdot \mathbf{u}^{n+1} |_{\partial\Omega}$$

and

$$\tau \cdot \mathbf{m}^{n+1} |_{\partial\Omega} = \tau \cdot (\mathbf{u}^{n+1} - \nabla \chi^{n+1}) |_{\partial\Omega}$$

This is almost identical to the boundary condition of \mathbf{u}^* in Alg 3. In normal mode analysis we show that a second order approximation to ϕ^{n+1} in the form of $\phi^{n+1} \simeq 2\phi^n - \phi$ along the boundary is necessary in obtaining second order accuracy in pressure.

Second step: imposing divergence free constraint:

$$\nabla^2 \chi^{n+1} = \nabla \cdot \mathbf{m}^{n+1}$$

with again a zero Neumann boundary condition in the normal component which is followed by the choice of boundary condition of \mathbf{m}^{n+1} specified in the previous step:

$$\mathbf{n} \cdot \nabla \chi^{n+1} |_{\partial\Omega} = \frac{\chi^{n+1}}{\mathbf{n}} |_{\partial\Omega} = 0$$

Third step: Then the velocities are updated with the formula:

$$\mathbf{u}^{n+1} = \mathbf{m}^{n+1} - \nabla \chi^{n+1}$$

Then pressure is updated as:

$$p^{n+1/2} = \frac{\chi^{n+1} - \chi^n}{\Delta t} - \frac{1}{2 Re} \nabla^2 (\chi^{n+1} + \chi^n) = \frac{\chi^{n+1} - \chi^n}{\Delta t} - \frac{1}{2 Re} \nabla \cdot (\mathbf{m}^{n+1} + \mathbf{m}^n)$$

Shen et.al has proven that through normal mode analysis in square domain and numerical results that only Gauge method shows fully second order Pressure error convergence in general domains. Later we demonstrate through numerical results that this is true. Recently Guermond and Shen have also proposed a fully second order accurate scheme called “Consistent” splitting method. They have shown that it is equivalent to the Gauge method [14, 24, 34]. However it is more suitable for finite element schemes whereas the original Gauge method we consider here is more designed for finite difference [24]

In this project the “Consistent splitting” method is not presented due to limited time and also because it is equivalent to Gauge method. There is no point of being duplicative here.

Chapter 4

Error Analysis of Projection methods

In this chapter we will present the error analysis for different “second order” Projection methods and the Gauge method through rigorous normal mode analysis.

4.1 Introduction of Normal mode analysis

There are many ways to analyse the convergence and error for numerical methods, including the popular Energy methods [14, 20] and Normal mode methods. Although the process of these methods differ but their essence is the same: compare the numerical solutions obtained to the true analytical solutions [15, 24]. Energy methods can be used in a general setting but depends on the particular error structure of the projection methods and sometimes difficult to obtain [14]. On the other hand, normal mode analysis is simple and reveals more precise information of the error of the projection methods. Hence recently there has been an increase of using normal mode analysis in the literature. However there is a major limitation because the analysis can only be performed on special domains and boundary conditions [6, 24, 28]. For instance, in Brown. et. al 2001 paper [6], only one normal mode was considered and hence the results could not be easily generalised.

Nevertheless, due to its simplicity and precision we will still perform a simple normal mode analysis to the “second order” projection methods in this section

4.2 Normal mode analysis of Projection methods

We are mainly concerned with the error in time stepping. The interaction of boundary conditions, pressure update and the second order implicit viscous term are more important to us. Thus the advective term can be neglected. [6, 15, 20, 24, 26–28]. This has been done in most of the textbooks and papers, and hence we will follow the convention and perform the analysis on the linearised unsteady Stokes equations.

We perform the analysis in a similar fashion to what has been done in David Brown paper [6] and work with a periodic semi-infinite channel which is one of the simplest settings to consider slip boundary conditions.

Periodic semi-infinite channel Errors in interior points can often easily be improved whereas it is more difficult to do so along the boundary. A physical boundary (e.g. no-slip boundary condition) could cause the error to behave differently. In most of the numerical studies (including this project) the flow is often confined in a bounded region, where slip-conditions are commonly used. For instance, a square domain with 4 sides. It is interesting to see how the Projection methods performs under these slip-boundary conditions. For the sake of simplicity we consider rather a microscopic picture of such domains where we focus on one side of domain only. A Dirichlet boundary condition is imposed on it whereas other parts are left free. This is like a 1-dimensional flow in a 2-dimensional plane. We achieve this by using a semi-infinite periodic channel where the x direction flow is non-trivial but the y direction flow is fixed to be periodic in time. This is essentially the same as the one used in Brown's paper [6]. While this geometry gives us an initial idea of the error behaviour for projection methods but it is too special and non-physical because we ignore the interaction between different boundaries. Whether error would behaves the same for other kinds of domains remains unanswered [24].

The domain is explicitly stated as:

$$\Omega = [0, \infty) \times [0, 2\pi], \quad t \in [0, T]$$

where T is an arbitrary end time.

and boundary points:

$$\partial\Omega = \{(x, y) \in \mathbb{R}^2 : x = 0\} \cup \{(x, y) \in \mathbb{R}^2 : y = 0, 2\pi\}$$

A Dirichlet boundary condition is imposed on the left end of the channel ($x = 0$) and the flow in y is assumed to be equal to the top and bottom boundaries to ensure periodicity ($\mathbf{u}(x, 0, t) = \mathbf{u}(x, 2\pi, t)$).

Let's first compute the reference solution.

The 2 dimensional $(u(x, y, t), v(x, y, t), p(x, y, t))$ initial value problem of the linearised Stokes equations is presented below. We assume $u(x, y, t), v(x, y, t), p(x, y, t)$ are solutions to the Stokes equations and they satisfy both the momentum and continuity equations.

$$\mathbf{u}_t + \nabla p = \frac{1}{R} \nabla^2 \mathbf{u}, \quad \text{in } \Omega \quad (4.1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega \quad (4.1b)$$

where R again denotes the Reynolds number

boundary conditions:

$$u(0, y, t) = \alpha(y, t), \quad \text{in } \partial\Omega \quad (4.1c)$$

$$v(0, y, t) = \beta(y, t), \quad \text{in } \partial\Omega \quad (4.1d)$$

initial value

$$\mathbf{u}(x, y, 0) = \mathbf{u}_0 \quad (4.1e)$$

For simplicity, let's assume $\mathbf{u}_0 = 0$

by taking the divergence to the momentum equation given above, we arrive at another important condition

$$\nabla^2 p = 0, \quad \text{in } \Omega \quad (4.1f)$$

Because of the periodic geometry, we are only concerned with the fluid motion in x direction. Thus we regard the flow as a “1-D” flow in x . We can then reduce the problem into ordinary differential equations (ODE) by taking Laplace transform in time and Fourier transform in the y direction. Taking the transform in x direction will yield the same result (of course this corresponds to periodic flow in x instead). Let s and k denote the Laplace and Fourier transform variables respectively.

The unilateral Laplace transform is given as

$$\mathcal{L}[f(t)] = \int_0^\infty f(t)e^{-st}dt$$

$f(t)$ is an integrable function defined over the interval $[0, \infty]$

and the standard Fourier transform (with angular frequency k) is defined as

$$\mathcal{F}[f(y)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(y)e^{-iky}dy$$

We take Laplace transform followed by Fourier transform to the linearised Stokes equations. Let a hat ($\hat{\cdot}$) denote the final transformed variable.

By the common Fourier and Laplace transform identities, we obtain the following transformed quantities.

$$\mathbf{u} \rightarrow \hat{\mathbf{u}} = (\hat{u}, \hat{v}), \quad p \rightarrow \hat{p}$$

$$\mathbf{u}_t \rightarrow s\hat{\mathbf{u}}, \quad \nabla \cdot \mathbf{u} \rightarrow \partial_x \hat{u} + ik\hat{v}$$

$$\nabla^2 \mathbf{u} \rightarrow (\partial_x^2 - k^2)\hat{\mathbf{u}}, \quad \nabla p \rightarrow (\partial_x p, ikp)^T$$

Hence the transformed Stokes equations are:

$$(\partial_x^2 - \mu^2)\hat{u} = R \partial_x \hat{p}, \quad \text{in } \Omega \quad (4.2a)$$

$$(\partial_x^2 - \mu^2)\hat{v} = ik R \hat{p}, \quad \text{in } \Omega \quad (4.2b)$$

$$(\partial_x^2 - k^2)\hat{p} = 0, \quad \text{in } \Omega \quad (4.2c)$$

$$\partial_x \hat{u} + ik\hat{v} = 0 \quad (4.2d)$$

$$\hat{u}(0, y, t) = \hat{\alpha}, \quad \text{in } \partial\Omega \quad (4.2e)$$

$$\hat{v}(0, y, t) = \hat{\beta}, \quad \text{in } \partial\Omega \quad (4.2f)$$

where $\mu^2 \equiv k^2 + R s$ and we take μ to be the positive real part of the solution for simplicity [6].

Because Equations (5.2) are ordinary differential equations, hence we can solve them easily using method of characteristics. Let's start from the pressure equation (E (4.2 c)).

Its characteristic equation is given as:

$$r^2 - k^2 = 0$$

Solving it we obtain:

$$r = \pm|k|$$

By superposition principle we obtain:

$$\hat{p} = P_1 e^{|k|x} + P_2 e^{-|k|x}$$

Hence there are 2 distinct modes of motion: exponential growth and decay. However because of our infinite half plane geometry, we would see the pressure (and also velocities) grow to infinity as $x \rightarrow \infty$. This is physically impossible and we don't want to see our solutions blow up in space! Therefore the exponential growth mode must be dropped, leaving the solution as:

$$\hat{p} = Pe^{-|k|x} \quad (4.3)$$

where P is a constant amplitude to be determined.

Hence we are left with only one normal mode. In fact, this exponential decaying motion is also referred as “quasi-normal mode” in literature.

Thus we have seen the limitation of normal mode analysis already because the normal modes strongly depends on the spatial domain of the problem.

Substitute the expression for \hat{p} into Equation 4.2 (a) we obtain an equation for \hat{u} :

$$\partial_x^2 \hat{u} - \mu^2 \hat{u} = -|k| Re Pe^{-|k|x} \quad (4.4a)$$

this is again an (inhomogeneous) ordinary differential equation for \hat{u} !

Solving it we obtain:

$$\hat{u} = U e^{-\mu x} + \frac{|k|}{s} P e^{-|k|x} \quad (4.4b)$$

where once again the exponentially growing mode ($e^{-\mu x}$) is dropped out.

Similarly by substituting the expression of pressure (Equation 4.3) into the ODE for \hat{v} (Equation 5.2 b) we obtain the solution for \hat{v} :

$$\hat{v} = V e^{-\mu x} - \frac{ik}{s} P e^{-|k|x} \quad (4.5)$$

And by the divergence constraint Equation 4.2 (d) we obtain another equation relating \hat{u} and \hat{v}

$$-\mu U + ik V = 0 \quad (4.6)$$

Now with 3 equations corresponding to 3 unknowns, we can solve for U, V and P by applying the boundary conditions given in Equation 4.2 (d-f). The problem is organised neatly in the following matrix form:

$$\begin{bmatrix} 1 & 0 & |k|/s \\ 0 & 1 & -ik/s \\ -\mu & ik & 0 \end{bmatrix} \begin{bmatrix} U \\ V \\ P \end{bmatrix} = \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \\ 0 \end{bmatrix} \quad (4.7)$$

Solving the matrix problem and we obtain the coefficients for the solutions as:

$$U = \frac{(\mu + |k|)}{Rs}(-|k|\hat{\alpha} + ik\hat{\beta}) \quad (4.8a)$$

$$V = \frac{-i(\mu + |k|)\mu}{Rs}(-\frac{k}{|k|}\hat{\alpha} + i\hat{\beta}) \quad (4.8b)$$

$$P = \frac{(\mu + |k|)}{R|k|}(\mu\hat{\alpha} - ik\hat{\beta}) \quad (4.8c)$$

And finally we arrive at the reference solutions for our initial value problem of the linearised Stokes equations:

$$\hat{u}_{ex} = \frac{(\mu + |k|)}{Rs}(-|k|\hat{\alpha} + ik\hat{\beta})e^{-\mu x} + \frac{(\mu + |k|)}{Rs}(\mu\hat{\alpha} - ik\hat{\beta})e^{-|k|x} \quad (4.9a)$$

$$\hat{v}_{ex} = \frac{-i(\mu + |k|)\mu}{Rs}(-\frac{k}{|k|}\hat{\alpha} + i\hat{\beta})e^{-\mu x} - \frac{ik(\mu + |k|)}{Rs|k|}(\mu\hat{\alpha} - ik\hat{\beta})e^{-|k|x} \quad (4.9b)$$

$$\hat{p}_{ex} = \frac{(\mu + |k|)}{R|k|}(\mu\hat{\alpha} - ik\hat{\beta})e^{-|k|x} \quad (4.9c)$$

where the subscript “ex” represents exact solutions.

4.2.1 Normal analysis for Algorithm 1,2,3

Now let's perform the above analysis to the numerical schemes described in the previous chapter.

Note that the numerical schemes we work with are only discretised in time because we are mainly concerned in the error behaviour in time stepping rather than spatial stepping. Therefore we will need to use discrete Laplace transform whereas keep using the continuous Fourier transform.

with a time step of Δt the discrete Laplace transform (or Z transform) is defined as [28]:

$$X(z) = \Delta t \sum_{n=-\infty}^{n=\infty} z^{-n} x(n), \quad z = e^{s\Delta t} \text{ and } |z| > 1$$

[6, 24, 28]

Note z is the discrete transform variable (not the z - direction) and it is related to the continuous transform by: $z = e^{s\Delta t}$.

The Z transform converts a series of discrete time signals $x(n)$ into complex frequencies domain. Hence X^{n+1} is merely X^n shifted up by z . To see this define

a new set of signals w being basically the original signal $x(n)$ at $n+1$ step. Hence $w^n = x^{n-1}$. Then by applying

$$\begin{aligned}
 X^{n+1} &= W^n \\
 &= \Delta t \sum_{\substack{n=\infty \\ n=-\infty}} w^n z^{-n} \\
 &= \Delta t \sum_{\substack{n=\infty \\ n=-\infty \\ n=\infty}} x^{n-1} z^{-n} \\
 &= \Delta t \sum_{\substack{n=\infty \\ n=-\infty}} x^n z^{-(n+1)} \\
 &= z \Delta t \sum_{\substack{n=\infty \\ n=-\infty}} x^n z^{-n} \\
 &= z X^n
 \end{aligned} \tag{4.10}$$

As discussed before there are 3 important things that are crucial to the accuracy of projection methods: q : choice of pressure approximation; boundary condition of the intermediate velocity field and choice of the non-physical variable ϕ resulted from projection. David has argued that the coupling between these 3 things must be considered to achieve second order accuracy for all primitive variables. We will also show this in the subsequent analysis [6]

Let's consider a general second order projection method with centred finite difference used to approximate the time derivative and 2nd order Crank-Nicholson method used to treat the implicit diffusive terms. This is just the linearised scheme considered in chapter 4.4.2 (Equation 4.23 to 4.25)

The algorithm is summarised below

$$\left\{
 \begin{array}{l}
 \frac{(\mathbf{u}^* - \mathbf{u}^n)}{\Delta t} + \nabla q = \frac{R}{2} \nabla^2 (\mathbf{u}^* + \mathbf{u}) \quad \text{solve for } \mathbf{u}^* \\
 \mathbf{u}^* |_{\partial\Omega} = (\mathbf{u}^{n+1} + \Delta t \phi^{n+1}) |_{\partial\Omega} \quad \text{Boundary condition and projection} \\
 p^{n+1/2} = q + L \phi^{n+1} \quad \text{Pressure update}
 \end{array}
 \right. \tag{4.11}$$

Where q , L , boundary conditions for projection and ϕ^{n+1} depends on the particular projection scheme.

Let's first consider the relation between q and ϕ^{n+1}

In Projection methods, the intermediate velocity field and auxiliary field are

approximations to the true velocity and pressure, although the degree of accuracy is dependent on different methods. Hence we expect a close relationship between ϕ and pressure. In fact, in order to eliminate the potential numerical boundary layers that might arise in computations, David has proposed a simple definition of ϕ :

Define:

$$q^n = Q(n)\phi^n$$

Apply Laplace and Fourier transform:

$$\hat{q}^n = Q(z)\hat{\phi}^n \quad (4.12)$$

Where $Q(z)$ is a function of Z transform variable that describes the coupling between the choice of pressure approximation (q) and the auxiliary field (ϕ). This function indeed varies for different variations of projection methods. We will see shortly that this limit the choice of q too.

In this section, we consider the 3 widely used Projection methods discussed in previous chapter.

Alg 1 - Projection with Lagged Pressure term (first order update formula)

As discussed in chapter 4.4.2 this method uses a first order pressure update formula:

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1}$$

Hence this method corresponds to

$$q = p^{n-1/2}, \quad L = I \text{ Identity matrix}$$

And boundary condition for projection:

$$\mathbf{u}^*|_{\partial\Omega} = \mathbf{u}^{n+1}|_{\partial\Omega}$$

Alg 2 - Projection with Lagged Pressure term (second order update formula)

This corresponds to a second order pressure update formula used to improve the accuracy for Alg 1

$$\begin{aligned} p^{n+1/2} &= p^{n-1/2} + \nabla\phi^{n+1} - \frac{\Delta t}{2R}\nabla^2\phi^{n+1} \\ q &= p^{n-1/2}, \quad L = I - \frac{\Delta t}{2R}\nabla^2 \end{aligned}$$

And same boundary condition for projection

Alg 3 - Pressure free projection First proposed by Kim and Moin. Pressure is can be recovered using the same second order update formula as in Alg 2. [6, 18]

$$q = 0, \quad L = I - \frac{\Delta t}{2R} \nabla^2$$

Boundary condition for projection:

$$\mathbf{n} \cdot \mathbf{u}^*|_{\partial\Omega} = \mathbf{n} \cdot \mathbf{u}^{n+1}|_{\partial\Omega}, \quad \tau \cdot \mathbf{u}^*|_{\partial\Omega} = \tau (\mathbf{u}^{n+1} + \Delta t \tau \nabla \phi^{n+1})|_{\partial\Omega}$$

Now Let's perform the Normal Mode analysis.

Taking divergence of Equation 4.11 (a) we can eliminate the divergence free velocity field. Note that the Laplace of \mathbf{u}^n is also divergence free because $\nabla \cdot \nabla^2 \mathbf{u}^n = \nabla^2 \nabla \cdot \mathbf{u}^n = 0$. Recall the definition of q in equation 4.12 we obtain an equation between \mathbf{u}^* and ϕ .

$$\nabla \cdot \mathbf{u}^* + \Delta t Q(n) \nabla^2 \phi^{n+1} = \frac{\Delta t}{2} \nabla^2 (\nabla \cdot \mathbf{u}^*) \quad (4.13)$$

Also taking divergence of the compatibility equation (Equation 4.11 (b)) and substitute into the equation above we obtain an equation in terms of $\nabla^2 \phi^{n+1}$, denote this as η

$$\eta + Q(n)\eta = \frac{\Delta t}{2} \nabla^2 \eta$$

The variable η describes the coupling between the intermediate velocity field \mathbf{u}^* and the scalar potential ϕ resulted from the projection. Therefore even though velocity and pressure are now decoupled from projection, their coupling is now represented by the intermediate velocity and the auxiliary variable ϕ . Hence it is important to understand the structure of η in order to understand the error in velocity and pressure for these projection methods.

Similar to the reference solution derivation, we take Laplace and Fourier transform to the equation above. After rearranging we obtain:

$$\partial_x^2 \hat{\eta} - \hat{\eta} (k^2 + \frac{2}{\Delta t} (1 + Q(z))) = 0 \quad (4.14)$$

Further define $\gamma^2 = k^2 + \frac{2}{\Delta t} F(z)$ where $F(z) = 1 + Q(z)$ for convenience and taking γ to be the positive real part of the previous equation.

By solving the ODE for $\hat{\eta}$ we found

$$\hat{\eta} = A e^{-\gamma x} \quad (4.15)$$

Note the non-physical exponential growing mode $e^{\gamma x}$ has already been dropped.

The decaying rate γ causes trouble because the Δt in the denominator of γ would cause the mode to become ill-defined as Δt approaching zero. This makes $A e^{-\gamma x}$ a spurious mode. [6, 28]. The numerical boundary layer and its effects on accuracy is well studied by the work of *E and Liu et al* [20].

It is worth noting that the spurious mode obtained by solving η comes from the compatibility equation (Equation 4.11 (b)). Hence it is an inherent property for all projection methods. Also recall the definition of η , it is customary to think the divergence of intermediate velocity ($\nabla \cdot \mathbf{u}^*$) and the auxiliary field (ϕ) would contain the spurious mode too. We will shortly demonstrate this claim. Therefore it is of interest to devise a method which ensures the actual velocity (\mathbf{u}) and pressure (p) does not contain it.

With the expression for η we can now solve for ϕ based on the second order inhomogeneous ordinary differential equation below:

$$(\partial_x^2 - k^2)\hat{\phi} = \eta = A e^{-\gamma x} \quad (4.16)$$

$$\hat{\phi} = \hat{\phi}_h + \hat{\phi}_p = A_1 e^{-|k|x} + A_2 e^{-\gamma x} \quad (4.17)$$

Where the subscript h and p denotes homogeneous and particular solutions from the ODE respectively.

We observe that ϕ_h contains the actual piece of solution we want whereas ϕ_p is not because it contains the spurious mode. Thus we have demonstrated our claim that the auxiliary variable ϕ contains the spurious mode. However as we shall see later that we do want $\nabla \cdot \mathbf{u}^*$ converges to zero as $\Delta t \rightarrow 0$ and then the homogeneous solution would dominate.

Hence we can now solve the intermediate velocity field by taking the divergence of compatibility condition:

$$\nabla \cdot \hat{\mathbf{u}}^* = \Delta t (\partial_x^2 - k^2)\hat{\phi} = \Delta t A_2 e^{-\gamma x} (\gamma^2 - k^2) \quad (4.18)$$

Because $\gamma^2 - k^2 = F(z) = 1 + Q(z)$ and this term obviously would not be zero for any of the projection methods, hence we have demonstrated that the divergence of the intermediate velocity field also contains the spurious mode. These findings are consistent with Brown, strikwerda and Shen [6, 28].

Now it is interesting to see how the spurious mode would affect the accuracy for velocity and pressure in the projection methods.

For Alg 1, the pressure update formula is:

$$p^{n+1/2} = q + \phi^{n+1} = (Q(n) + 1)\phi^{n+1} \quad (4.19)$$

Where the lagged pressure approximation $q = p^{n-1/2}$ is used.

$$\hat{p}^{n+1/2} = (Q(z) + 1)\hat{\phi} \quad (4.20)$$

Then by recalling the relation between q and ϕ (Equation 5.12) we found that:

$$\hat{q} = Q(z)\hat{\phi} = \hat{p}^{n-1/2} = \frac{1}{z^{3/2}}\hat{p} = \frac{1}{z^{3/2}}z^{1/2}(Q(z) + 1)\hat{\phi}$$

Then we obtain

$$Q(z) = \frac{1}{z}(Q(z) + 1)$$

This finally leads to the following expression of $Q(z)$:

$$Q(z) = \frac{1}{z - 1} \quad (4.21)$$

Hence we found that the pressure is updated as:

$$\begin{aligned} \hat{p}^{n+1/2} &= \frac{1}{z - 1} (A_1 e^{-|k|x} \\ &\quad + A_2 e^{-\gamma x}) + A_1 e^{-|k|x} + A_2 e^{-\gamma x} = \frac{z}{z - 1} (A_1 e^{-|k|x} + A_2 e^{-\gamma x}) \end{aligned} \quad (4.22)$$

The clearly indicates the presence of spurious mode. Hence the the accuracy is strongly limited due to formation of numerical boundary layer. The boundary layer not only causing degradation in accuracy along the boundary but could also affect the nearby interior points too. In some domains (like a squared domain with Drichlet boundary condition imposed), the spurious mode would cause strong oscillations along the boundary.

In addition as the projection implies, the auxiliary field often satisfies a normal Neumann boundary condition :

$$\mathbf{n} \cdot \nabla \phi^{n+1} = \frac{\partial \phi^{n+1}}{\partial n} = 0$$

This is often referred as a non-physical boundary condition in the literature [6, 15, 28] for the reasoning discussed below.

Extracting the normal component of the gradient of pressure along the boundary (Equation 5.19) we found:

$$\mathbf{n} \cdot \nabla \hat{p}^{n+1/2} |_{\partial\Omega} = \mathbf{n} \cdot \nabla \hat{p}^{n-1/2} |_{\partial\Omega} \quad (4.23)$$

This implies the normal pressure gradient is constant along the boundary for all time steps. This is in general not consistent with the actual boundary condition of pressure gradient. Hence this partly explains why numerical boundary layer exists.

The other update formula used for Alg 2 and 3 should show an improved accuracy.

The transformed version of the formula is:

$$\hat{p}^{n+1/2} = \frac{1}{z^{1/2}} \hat{p} = Q(z) \hat{\phi} + \hat{\phi} - \frac{1}{2} \nabla \cdot \hat{\mathbf{u}}^* \quad (4.24)$$

substitute the expression for $\nabla \cdot \hat{\mathbf{u}}^*$ and $\hat{\phi}$ into the equation above we obtain:

$$\frac{1}{z^{1/2}} \hat{p} = Q(z) A_1 e^{-|k|x} + Q(z) A_2 e^{-\gamma x} + A_1 e^{-|k|x} + A_2 e^{-\gamma x} - (1 + Q(z)) e^{-\gamma x} \quad (4.25)$$

After rearranging we found that the spurious mode $e^{-\gamma x}$ is actually being cancelled out, leaving the pressure as:

$$\hat{p} = z^{1/2} (Q(z) + 1) A_1 e^{-|k|x} \quad (4.26)$$

where the $Q(z)$ varies depending on q . It could be $\frac{1}{z-1}$ as in Alg 1 and Alg 2 or zero in Alg 3. However in either case, the spurious is filtered out from the pressure update equation. Here from this point of view, this formula shows an improved accuracy.

However this analysis needs more attention for Alg 2, because the pressure approximation uses a lagged pressure value and recall the relation between q and ϕ we find:

$$p^{n-1/2} = Q(n) \phi^{n+1}$$

Because ϕ^{n+1} satisfy a zero Neumann boundary condition hence the lagged pressure has a zero normal gradient.

$$\mathbf{n} \cdot \nabla p^{n-1/2} = \mathbf{n} \cdot Q(n) \nabla \phi^{n+1} \quad (4.27)$$

This seems like to be a contradictory observation as the normal pressure gradient is not zero at $n+1/2$ step as indicated by the new update formula. Hence it

seems that this is enforcing the newly calculated pressure gradient to be zero after each iteration. Therefore we infer that this choice of q would result in a poorer approximation to the true normal pressure gradient especially where the analytical counterpart is not zero along the boundary. In practice, the normal pressure approximation is made consistent across all iterations because we don't force a change on $p^{n-1/2}$. However this implies the relation $q = p^{n-1/2} = Q(n)\phi^{n+1}$ does not hold any more. Hence the numerical boundary layer actually would not be filtered out completely with this choice of q . Later we will see an illustration of this problem. Alg 3 does not suffer from this problem as $Q = 0$ (no pressure approximation is used at all). Therefore the normal mode analysis indicates appropriate choice of pressure approximation is critical to the accuracy of projection methods. A modification of q for Alg 2 is proposed and discussed in the results section.

Now let's solve for the velocities.

First substitute the compatibility equation (4.11 (b)) into the momentum equation (4.11 (a)) to eliminate \mathbf{u}^* and then taking Laplace and Fourier transform in y to the resulting equation.

Also recall $q = Q(n)\phi^{n+1}$, $\hat{u}^{n+1} = z\hat{u}^n$, $\hat{\phi}^{n+1} = z\hat{\phi}^n$ and dropping the index $n + 1$ we obtain

$$(\partial_x^2 - \bar{\mu}^2)\hat{\mathbf{u}} = \frac{z\Delta t}{z+1} [-(\partial_x^2 - \lambda^2) + \frac{2RQ(z)}{\Delta t}] \nabla \hat{\phi} \quad (4.28)$$

where again we have defined $\bar{\mu}$ to be the positive real part of $\bar{\mu}^2 = k^2 + R\rho$, $\rho = \frac{2(z-1)}{\Delta t(z+1)}$ and $\lambda^2 = k^2 + \frac{2R}{\Delta t}$ [6].

With the solution of $\hat{\phi}$ obtained earlier (equation 5.17) we can then solve for velocities accordingly

$$(\partial_x^2 - \bar{\mu}^2)\hat{\mathbf{u}} = \frac{z\Delta t}{z+1} [-(\partial_x^2 - \lambda^2) + \frac{2RQ(z)}{\Delta t}] \nabla (\hat{\phi}_h + \hat{\phi}_p) \quad (4.29)$$

Expanding the right hand side and using the definition of γ^2 and λ^2 :

$$\frac{z\Delta t}{z+1} [-(\partial_x^2 - k^2) \nabla \hat{\phi}_h + \frac{2R}{\Delta t} (1 + Q(z)) \nabla \hat{\phi}_h - \partial_x^2 \nabla \hat{\phi}_p + \gamma^2 \nabla \hat{\phi}_p]$$

It is worth to note that most of the terms above can actually be dropped out including $\hat{\phi}_p$ which contains the spurious mode.

To see this recall $(\partial_x^2 - k^2)\hat{\phi} = \hat{\eta}$ and hence the gradient of $\hat{\phi}$ satisfies the equation

too by an interchange of operators. Therefore $(\partial_x^2 - k^2)\nabla\hat{\phi}_h = 0$; now recall the particular solution of $\hat{\phi}$ is: $A_2 e^{-\gamma x}$ hence we have:

$$-\partial_x^2 \nabla \hat{\phi}_p + \gamma^2 \nabla \hat{\phi}_p = \begin{cases} -\partial_x^2 \partial_x (A_2 e^{-\gamma x}) + \gamma^2 \partial_x (A_2 e^{-\gamma x}) = \gamma^3 A_2 e^{-\gamma x} - \gamma^3 A_2 e^{-\gamma x} = 0 \\ -\partial_x^2 i j (A_2 e^{-\gamma x}) + \gamma^2 i k (A_2 e^{-\gamma x}) = -\gamma^2 i k A_2 e^{-\gamma x} + \gamma^2 i k A_2 e^{-\gamma x} = 0 \end{cases}$$

These results indicates that the spurious mode contained in $\hat{\phi}_p$ is now eliminated from the right hand side of equation 4.29, leaving a simpler equation to solve for $\hat{\mathbf{u}}$:

$$(\partial_x^2 - \bar{\mu}^2) \hat{\mathbf{u}} = \frac{2z(1+Q(z))}{z+1} \nabla \hat{\phi}_h \quad (4.30)$$

For \hat{u} component:

$$(\partial_x^2 - \bar{\mu}^2) \hat{u} = -\frac{2z(1+Q(z))}{z+1} |k| A_1 e^{-|k|x}$$

Solving this inhomogeneous ordinary differential equation we obtain

$$\hat{u} = U e^{-\bar{\mu}x} + \frac{G(z)}{\rho} |k| A_1 e^{-|k|x}$$

where for convenience we have defined $G(z) = \frac{2zF}{1+z}$

Similar process can be used to solve for \hat{v} .

In summary we have:

$$\begin{cases} \hat{u}_{nu} = U e^{-\bar{\mu}x} + \frac{G(z)}{\rho} |k| A_1 e^{-|k|x} \\ \hat{v}_{nu} = V e^{-\bar{\mu}x} - \frac{G(z)}{\rho} i k A_1 e^{-|k|x} \\ \hat{\phi}_{nu} = A_1 e^{-|k|x} + A_2 e^{-\gamma x} \end{cases} \quad (4.31)$$

where “nu” refers to numerical solutions.

Now we can apply the boundary conditions to solve for the undetermined coefficients.

Recall at $x = 0$, the exact solutions should satisfy: (where $t_n = (n+1)\Delta t$)

$$u(0, y, t_n) = \alpha, \quad v(0, y, t_n) = \beta, \quad (4.32a)$$

$$\partial_x u(0, y, t_n) + \partial_y v(0, y, t_n) = 0$$

As implied by the projection the numerical solutions should satisfy

$$\begin{aligned} u^*|_{x=0} &= u^{n+1}|_{x=0} = \alpha, \quad \phi_x^{n+1}|_{x=0} = 0, \\ v^*|_{x=0} &= (v^{n+1} - \Delta t \partial_y \phi^{n+1})|_{x=0} \end{aligned} \quad (4.32b)$$

The tangential boundary condition needs more attention because we actually don't have access to ϕ^{n+1} when solving for \mathbf{u}^* . Hence approximation is needed. Introducing function $\mathcal{B}(\phi)$ which approximates ϕ^{n+1} (\mathcal{B} depends on the particular projection methods). In practice, this implies the following tangential boundary condition for intermediate velocity:

$$v^*|_{x=0} = (\beta - \Delta t \partial_y \mathcal{B}(\phi))|_{x=0} \quad (4.32c)$$

It is important to point out that now the numerical velocity v^{n+1} does not equal to the exact $v(t_n)$ along the boundary any more (except we enforce $\phi^{n+1} = 0$ along the boundary). It rather approximates $v(t_n)$ to an order of accuracy depends on function \mathcal{B} . This is a well known phenomenon in projection method [6, 28]. 3 choices of \mathcal{B} is made here with $\mathcal{B} = 0, \phi^n, 2\phi^n - \phi^{n-1}$ which corresponds to no approximation, first order and second order approximations to ϕ^{n+1} . We will see that choice of the approximations has a critical impact on the accuracy of the projection methods.

Combine Equation 4.32c and 4.32 b and performing Fourier transformations to the boundary conditions we obtain

$$\hat{u} = \hat{\alpha} \quad (4.32d)$$

$$(\hat{v}^{n+1} + ik\Delta t(\hat{B} - 1)\hat{\phi})|_{x=0} = \hat{\beta} \quad (4.32e)$$

where we have introduced $\hat{B}\hat{\phi} = \hat{\mathcal{B}}$

$$\partial_x \hat{\phi}|_{x=0} = 0 \quad (4.32f)$$

Now let's compute the coefficients first

From Equation 4.32 and the expression of \hat{u} and \hat{v} we obtain the following relations:

$$U = \hat{\alpha} - \frac{G(z)|k|}{\rho} A_1 \quad (4.33a)$$

$$\bar{\mu}U = ikV \quad (4.33b)$$

$$-|k|A_1 - \gamma A_2 = 0 \Rightarrow A_2 = -\frac{|k|}{\gamma} A_1 \quad (4.33c)$$

$$V - \frac{ikG(z)}{\rho}A_1 + ik\Delta t(\hat{B} - 1)(A_1 + A_2) = \hat{\beta} \quad (4.33d)$$

We can then solve for A_1 by manipulating the above equations and the expressions for U and V . Also need to use the relation introduced before: $\Delta t = \frac{2F}{\gamma^2 - k^2}$. After some tedious algebra we obtain:

$$A_1 = E^{-1}(\bar{\mu}\hat{\alpha} - ik\hat{\beta}) \quad (4.34)$$

where we have defined: $E = \frac{RG(z)|k|}{\bar{\mu} + |k|}(1 + \frac{CF(\hat{B} - 1)}{G(z)})$ and $C = \frac{2|k|(\bar{\mu} + |k|)}{\gamma(\gamma + |k|)}$ for convenience.

E represents precisely the coupling between the choice of pressure approximation ($G(z)$ and F) and choice of boundary condition of projection (this will affect \hat{B}). The coupling between these functions must be carefully chosen to maintain second order accuracy for all variables across the domain. We will see in the subsequent discussions how this can be met.

Then substitute the expression for A_1 back into Equations 4.33 and we recover other coefficients too:

In summary:

$$\begin{cases} A_1 = \frac{(\bar{\mu} + |k|)(\bar{\mu}\hat{\alpha} - ik\hat{\beta})}{RG(z)|k|}(1 + \frac{CF(\hat{B} - 1)}{G(z)})^{-1} \\ U = \hat{\alpha} - \frac{(\bar{\mu}\hat{\alpha} - ik\hat{\beta})(\bar{\mu} + |k|)}{R\rho}(1 + \frac{CF(\hat{B} - 1)}{G(z)})^{-1} \\ A_2 = -\frac{(\bar{\mu}\hat{\alpha} - ik\hat{\beta})(\bar{\mu} + |k|)}{\gamma R\rho G(z)}(1 + \frac{CF(\hat{B} - 1)}{G(z)})^{-1} \\ V = \frac{\bar{\mu}\hat{\alpha}}{ik} - \frac{\bar{\mu}(\bar{\mu}\hat{\alpha} - ik\hat{\beta})(\bar{\mu} + |k|)}{ik R\rho}(1 + \frac{CF(\hat{B} - 1)}{G(z)})^{-1} \end{cases} \quad (4.35)$$

For our purpose of testing the accuracy, we want to compare the expression between the reference solutions (Equation 4.9) and our numerical solutions (Equation 4.31).

Error analysis of velocity For \hat{u} velocity, let's examine the accuracy of the particular solution component because both the numerical and the reference components have the same decaying rate ($-|k|$).

Thus we compare

$$\frac{G(z)|k|}{\rho} A_1 \text{ numerical with } \frac{(\mu + |k|)}{R s} (\mu \hat{\alpha} - ik \hat{\beta}) \text{ reference solution}$$

Followed from the result for A_1

$$\frac{G(z)|k|}{\rho} A_1 = \frac{G(z)|k|}{\rho} E^{-1} (\bar{\mu} \hat{\alpha} - ik \hat{\beta})$$

Therefore it is obvious that we want

$$\frac{G(z)|k|}{\rho} E^{-1} = \frac{(\mu + |k|)}{R s} + \mathcal{O}(\Delta t^2) \text{ and } (\bar{\mu} \hat{\alpha} - ik \hat{\beta}) = (\mu \hat{\alpha} - ik \hat{\beta}) + \mathcal{O}(\Delta t^2) \quad (4.36)$$

where we have used the “Big \mathcal{O} ” notation to express the error.

Let's do the second one first since it is easier! Ideally we want

$$\mu \hat{\alpha} - ik \hat{\beta} = \bar{\mu} \hat{\alpha} - ik \hat{\beta} + \mathcal{O}(\Delta t^2)$$

Observe that the only term inhibits the accuracy is $\bar{\mu}$.

Recall $\bar{\mu}^2 = k^2 + R \rho$ and $\mu^2 = k^2 + R s$ Hence it is ρ which produces the error. A rough estimate indicate that we want ρ converging to s at least with second order accuracy. Hence let's prove this hypothesis first!

Recall the definition of ρ :

$$\rho = \frac{2(z-1)}{\Delta t(z+1)}$$

because by definition, $z = e^{s\Delta t} = \sum_{n=0}^{\infty} \frac{(s\Delta t)^n}{n!} = 1 + \mathcal{O}(\Delta t)$, hence $(z-1)^n = \mathcal{O}(\Delta t^n)$

By Taylor expansion of $f(z) = \frac{z-1}{z+1}$ at $z=1$ to order $(z-1)^3$ we obtain:

$$\begin{aligned} f(z) &= \sum_{n=0}^{\infty} \frac{f^{(n)}(1)(z-1)^n}{n!} \\ &= f(1) + f'(1)(z-1) + \frac{f''(1)(z-1)^2}{2} + \mathcal{O}(\Delta t^3) \\ &= \frac{e^{s\Delta t} - 1}{2} - \frac{(e^{s\Delta t} - 1)^2}{4} \end{aligned} \quad (4.37a)$$

By expanding $e^{s\Delta t}$ using the definition we find

$$f(z) = \frac{s\Delta t}{2} - \frac{(s\Delta t)^3}{4} + \mathcal{O}(\Delta t^3) \quad (4.37b)$$

Then it is easy to show that

$$\begin{aligned} \rho &= \frac{2f(z)}{\Delta t} \\ &= s - \frac{s^3 \Delta t^2}{2} + \mathcal{O}(\Delta t^2) \\ &= s + \mathcal{O}(\Delta t^2) \end{aligned} \quad (4.37c)$$

As for the error between $\bar{\mu}$ and μ , it is straightforward to use a similar Taylor series argument and the result proven for ρ to show

$$\bar{\mu} = \mu + \mathcal{O}(\Delta t^2) \quad (4.38)$$

These are the two important relations that we use in the accuracy test.

Now for the first comparison (in Equation 4.36), there are three varying functions C , $G(z)$ and $B(z)$ that we must consider to obtain second order accuracy.

$$G(z)|k|E^{-1} = \frac{\bar{\mu} + |k|}{R} \left(1 + \frac{C F(\hat{B} - 1)}{G(z)}\right)^{-1} \text{ compare with } \frac{\mu + |k|}{R}$$

Hence according to the above formulation, we want $\left(1 + \frac{C F(\hat{B} - 1)}{G(z)}\right)$ to converge to 1 (equivalently $\frac{C F(\hat{B} - 1)}{G(z)}$ to 0) at second order rate. By varying the pressure approximation function we can make this possible. Recall the 3 choices of \mathcal{B} for projection methods

$$\begin{aligned} \mathcal{B} = 0 &\Rightarrow \hat{B} = 0 && \text{no approximation to } \phi^{n+1} \text{ Alg 1 and Alg 2} \\ \mathcal{B} = \phi^n &\Rightarrow \hat{B} = \frac{1}{z} && \text{first order approximation to } \phi^{n+1} \text{ Alg 3} \\ \mathcal{B} = 2\phi^n - \phi^{n-1} &\Rightarrow \hat{B} = \frac{2}{z} - \frac{1}{z^2} && \text{second order approximation Alg 3} \end{aligned} \quad (4.39)$$

Let's consider these methods separately here:

For Alg 1 and Alg 2 with $\mathcal{B} = 0$, we have:

$$\left| \frac{C F(\hat{B} - 1)}{G(z)} \right| = \left| \frac{C F}{G(z)} \right|$$

Substitute the expression for C and using the identity: $|\gamma(\gamma + |k|)| \geq |\gamma^2|$ we obtain:

$$\left| \frac{C F}{G(z)} \right| \leq \left| \frac{2|k|(\bar{\mu} + |k|)F}{\gamma^2 G(z)} \right|$$

Recall the definition of γ^2 the above inequality becomes:

$$\left| \frac{2|k|(\bar{\mu} + |k|)F}{\left(k^2 + \frac{2F}{\Delta t} \right) G(z)} \right| \leq \left| \frac{|k|(\bar{\mu} + |k|)}{\left(\frac{G(z)}{\Delta t} \right)} \right| = \left| \frac{\Delta t |k|(\bar{\mu} + |k|)}{G(z)} \right|$$

For Alg 1 and 2 we know that $Q(z) = \frac{1}{z-1}$ and so

$$G(z) = \frac{2z(1+Q(z))}{z+1} = \frac{2z^2}{z^2-1}$$

Then it is quite straightforward to show that

$$\frac{1}{G(z)} = \frac{z^2-1}{2z^2} = \frac{1}{2} - \frac{1}{2}z^{-2} = \mathcal{O}(\Delta t)$$

Then we finally have that:

$$\left| \frac{C F}{G(z)} \right| \leq \left| \frac{\Delta t |k|(\bar{\mu} + |k|)}{G(z)} \right| = \mathcal{O}(\Delta t^2) \quad (4.40)$$

In addition since we have just proved $\bar{\mu} = \mu + \mathcal{O}(\Delta t^2)$, hence $G(z)|k|E^{-1}$ is a second order approximation to $\frac{\mu + |k|}{R}$. Therefore the boundary condition: $\mathbf{u}^*|_{x=0} = \mathbf{u}^{n+1}|_{x=0} = (\alpha, \beta)$ does give second order accuracy for the coefficients of the numerical solutions.

For Alg 3, the problem is more complicated because now the extra term $\frac{C F(\hat{B} - 1)}{G(z)}$ in the coefficient A_1, A_2 only converge to 0 at first order rate.

Similar to the analysis done for Alg 2 and 3 we know that:

$$\left| \frac{C F}{G(z)} \right| \leq \left| \frac{\Delta t |k|(\bar{\mu} + |k|)}{G(z)} \right|$$

Because now $Q(z) = 0$ since $q = 0$ and so $G(z) = \frac{2z}{z+1}$. This implies

$$\frac{1}{G(z)} = \frac{1+z}{2z} = \frac{1}{2} + \frac{1}{2}z = 1 + \mathcal{O}(\Delta t)$$

Hence we find that

$$\left| \frac{C F}{G(z)} \right| \leq \left| \frac{\Delta t |k| (\bar{\mu} + |k|)}{G(z)} \right| = \mathcal{O}(\Delta t) \quad (4.41)$$

which is only first order! This makes $G(z)|k|E^{-1} = \frac{\bar{\mu} + |k|}{R}(1 + \frac{C F(\hat{B} - 1)}{G(z)})^{-1} = \frac{\mu + |k|}{R} + \mathcal{O}(\Delta t)$ which will also make the U velocity only first order accurate.

This implies that we need more accurate approximation to ϕ^{n+1} along the boundary. Then consider a first order approximation corresponding to $\hat{B} = \frac{1}{z}$.

Then the extra term:

$$\left| \frac{C F \hat{B} - 1}{G(z)} \right| \leq \left| \frac{C F}{G(z)} \right| |\hat{B} - 1|$$

Now we want $\hat{B} - 1$ to be first order accurate at least. This is indeed true since:

$$\hat{B} - 1 = z^{-1} - 1 = \mathcal{O}(\Delta t) \quad (4.42)$$

Similarly for $\mathcal{B} = 2\phi^n - \phi^{n-1}$ which is a second order approximation to ϕ^{n+1} would result in $\hat{B} - 1 = \mathcal{O}(\Delta t^2)$, however first order accuracy is enough here.

Combine this we obtain the desired result:

$$\left| \frac{C F (\hat{B} - 1)}{G(z)} \right| = \mathcal{O}(\Delta t^2) \quad (4.43)$$

Hence for Alg 1, 2 and 3 we find that the following results hold:

$$\frac{G(z)|k|}{\rho} A_1 = \frac{(\mu + |k|)}{Rs} (\mu \hat{\alpha} - ik \hat{\beta}) + \mathcal{O}(\Delta t^2) \quad (4.44)$$

Now we can finally examine the accuracy for the first component of \hat{u} : compare $Ue^{-\bar{\mu}x}$ numerical and $\frac{(\mu + |k|)}{Rs} (-|k|\hat{\alpha} + ik\hat{\beta})e^{-\mu x}$ reference solution.

First note since $\bar{\mu} = \mu + \mathcal{O}(\Delta t^2)$ then

$$\begin{aligned} |e^{\bar{\mu}x} - e^{\mu x}| &= \left| \sum_{n=0}^{\infty} \frac{(\bar{\mu}x)^n - (\mu x)^n}{n!} \right| \\ &= |0 + (\bar{\mu}x - \mu x) + \frac{((\bar{\mu}x)^2 - (\mu x)^2)}{2} + \dots \\ &\leq |\bar{\mu} - \mu||x| + \left| \frac{((\bar{\mu}x)^2 - (\mu x)^2)}{2} \right| + \dots \end{aligned}$$

Neglecting higher error terms we obtain

$$|e^{\bar{\mu}x} - e^{\mu x}| \leq \mathcal{O}(\Delta t^2)$$

As for their coefficients

$$U = \hat{\alpha} - \frac{(\bar{\mu}\hat{\alpha} - ik\hat{\beta})(\bar{\mu} + |k|)}{Re \rho} \left(1 + \frac{CF(\hat{B} - 1)}{G(z)}\right)^{-1} = \frac{(\bar{\mu} + |k|)(-|k|\hat{\alpha} + ik\hat{\beta})}{Re \rho} \left(1 + \frac{CF(\hat{B} - 1)}{G(z)}\right)^{-1}$$

Since we have shown that $(1 + \frac{CF(\hat{B} - 1)}{G(z)})^{-1} = 1 + \mathcal{O}(\Delta t^2)$ holds for all projection methods as long as appropriate boundary conditions are chosen, then we find that

$$U = \frac{(\bar{\mu} + |k|)(-|k|\hat{\alpha} + ik\hat{\beta})}{R \rho} = \frac{(\mu + |k|)(-|k|\hat{\alpha} + ik\hat{\beta})}{Rs} + \mathcal{O}(\Delta t^2) \quad (4.45)$$

Further combine the above equation with Equation 4.44 and compare with the reference solution in Equation 4.9 (a) we find that

$$\begin{aligned} \hat{u}_{nu} &= U e^{-\bar{\mu}x} + \frac{G(z)}{\rho} |k| A_1 e^{-|k|x} \\ &= \frac{(\mu + |k|)(-|k|\hat{\alpha} + ik\hat{\beta})}{Rs} e^{-\mu x} + \frac{(\mu + |k|)(\mu\hat{\alpha} - ik\hat{\beta})}{Rs} e^{-|k|x} \\ &= \hat{u}_{ex} + \mathcal{O}(\Delta t^2) \end{aligned}$$

The error analysis for pressure is more complicated since the accuracy vary on different projection methods.

Let's consider the 3 projection methods separately.

First for Alg 1

This corresponds to $q = p^{n-1/2}$, $L = I$ and $\mathcal{B} = 0$ ($\hat{B} = 1$). The boundary

condition for \mathbf{u}^{n+1} is satisfied exactly leading to $\mathbf{u}^*|_{x=0} = \mathbf{u}^{n+1}|_{x=0} = (\alpha((n+1)\Delta t), \beta((n+1)\Delta t))$.

Recall the transformed pressure update formula derived in Equation 5.22 we have:

$$\hat{p} = z^{1/2} \hat{p}^{n+1/2} = \frac{z^{3/2}}{z-1} A_1 e^{-|k|x} + \frac{z^{3/2}}{z-1} A_2 e^{-\gamma x}) \quad (4.46)$$

It is obvious that this formula would result in a degraded accuracy because the second term $\frac{z^{3/2}}{z-1} A_2 e^{-\gamma x})$ contains the spurious mode. It has the coefficient written explicitly as:

$$\frac{z^{3/2}}{z-1} A_2 = -\frac{z^{3/2}}{z-1} \frac{(\bar{\mu}\hat{\alpha} - ik\hat{\beta})(\bar{\mu} + |k|)}{\gamma R\rho G(z)}$$

Clearly this term is not zero. In fact by using a Taylor series expansion at $z = 1$ we find that $\frac{z^{3/2}}{z-1} A_2 = \mathcal{O}(\Delta t)$ and hence pressure will actually be first order accurate.

Second for Alg 2. The pressure approximation and boundary condition for \mathbf{u}^* are the same with Alg 1 however in this case we are using the more accurate pressure update formula.

$$\hat{p} = z^{1/2} (Q(z) + 1) A_1 e^{-|k|x} \quad (4.47)$$

Also being $Q(z) = \frac{1}{z}$ the same as Alg 1, we have $G(z) = \frac{2z(1+Q(z))}{1+z} = \frac{2z^2}{z^2-1}$.

Combine with the expression of A_1 in Equation 4.37 (a) we obtain:

$$\hat{p} = \frac{z+1}{2z^{1/2}} \frac{(\bar{\mu} + |k|)(\bar{\mu}\hat{\alpha} - ik\hat{\beta})}{R|k|} (1 - \frac{CF}{G(z)})^{-1} e^{-|k|x} \quad (4.48)$$

and then recall the reference pressure solution in Equation 4.9 (c)

$$|\hat{p}_{nu} - \hat{p}_{ex}| = \left| \frac{z+1}{2z^{1/2}} \frac{(\bar{\mu} + |k|)(\bar{\mu}\hat{\alpha} - ik\hat{\beta})}{R|k|} (1 - \frac{CF}{G(z)})^{-1} - \frac{(\mu + |k|)(\mu\hat{\alpha} - ik\hat{\beta})}{R|k|} \right|$$

By using a Taylor series expansion we find that $\frac{z+1}{2z^{1/2}} = 1 + \mathcal{O}(\Delta t^2)$ and combine the result that $\bar{\mu} = \mu + \mathcal{O}(\Delta t^2)$, it is straightforward to show that $\hat{p}_{nu} = \hat{p}_{ex} + \mathcal{O}(\Delta t^2)$

Hence here we have proved that by eliminating the spurious mode, the new pressure update formula used in Alg 2 indeed provides second order accuracy in pressure.

For Alg 3, the same modified pressure update formula is used (as in Alg 2), so the spurious mode is again eliminated. The pressure approximation is now $q = 0$ leads to $Q(z) = 0$. The transformed version of pressure update formula therefore is:

$$\hat{p} = z^{1/2} A_1 e^{-|k|x}$$

We want the coefficient A_1 be at least second order accurate compared to the coefficient for the reference pressure solution. If $\hat{B} = 1$ as in Alg 1 and 2, this leads to $\left(1 + \frac{C F(\hat{B} - 1)}{G(z)}\right) = 1 + \mathcal{O}(\Delta t)$ and eventually lead to a first order accuracy in numerical pressure.

Thus considering more accurate boundary conditions, for instance: $\hat{B} = \frac{1}{z}$ then $\left(1 + \frac{C F(\hat{B} - 1)}{G(z)}\right) = 1 + \mathcal{O}(\Delta t^2)$. Then

$$\hat{p} = \frac{z+1}{2z^{1/2}} \frac{(\bar{\mu} + |k|)(\bar{\mu}\hat{\alpha} - ik\hat{\beta})}{R|k|} \left(1 + \frac{C F(\hat{B} - 1)}{G(z)}\right)^{-1} e^{-|k|x}$$

which then clearly gives second order accuracy.

Same result holds if second order accurate approximation to ϕ^{n+1} which corresponds to $\hat{B} = \frac{2}{z} - \frac{1}{z^2}$.

Hence our analysis show that Alg 3 (the pressure free projection method) needs at least first order approximation of ϕ^{n+1} when computing \mathbf{u}^* along the boundary. This is in agreement with the findings in literature [6, 28]. In addition, this is also supported by our numerical result.

4.2.2 Normal mode analysis of Gauge method

In this subsection, the accuracy of the Gauge method is explored. We will demonstrate that the introduction consistent update of Gauge variable lead to significant improvement over the accuracy in pressure simulations.

For simplicity let's take the periodic channel analysed for Projection methods before. The geometry is the same and hence we take the same boundary conditions for analytical velocities (\mathbf{u}).

Recall the semi-discretised Gauge formulation for the linearised Stokes equations:

$$\frac{\mathbf{m}^{n+1} - \mathbf{m}^n}{\Delta t} = \frac{1}{2} \nabla^2 (\mathbf{m}^{n+1} + \mathbf{m}^n) \quad (4.49a)$$

By the periodic geometry, the boundaries at $y = 0, 2\pi$ are periodic. Only the Dirichlet boundary at $x = 0$ needs to be specified. At this boundary, the x direction is normal and the y direction is tangential. Hence we have

$$u(0, y, t) = \alpha(y, t), \quad \text{and } v(0, y, t) = \beta(y, t) \quad (4.49b)$$

$$m_1(0, y, t) = \alpha(y, t), \quad \text{and } m_2(0, y, t) = v(0, y, t) + \partial_y \chi(0, y, t) \quad (4.49c)$$

Performing projection

$$\nabla^2 \chi^{n+1} = \nabla \cdot \mathbf{m}^{n+1}, \quad \text{with } \partial_x \chi^{n+1} = 0 \quad (4.49d)$$

$$\mathbf{u}^{n+1} = \mathbf{m}^{n+1} - \nabla \chi^{n+1} \quad (4.49e)$$

By making the normal boundary condition the same for the Gauge variable and velocity \mathbf{u}^* , we obtain a zero Neumann boundary condition for the auxiliary field χ^{n+1} . Similar to the projection method, this is one of the most common boundary condition for auxiliary fields.

Again the Reynolds number is made to be 1 for simplicity.

Because the Gauge variable \mathbf{m} is being updated consistently, hence like velocities and pressure we have a nice relationship between the Gauge variables calculated at different time iterations: $\mathbf{m}^{n+1} = z\mathbf{m}^n$ (where z again is the discrete Laplace transform variable). This was not possible for the intermediate velocity field shown in the projection methods because it is rather discarded after each iteration. We cannot really locate the variable in time and thus makes the tracking of its errors difficult.

In this case, we can solve the Gauge variables directly by performing Laplace and Fourier transforms.

$$\frac{\hat{\mathbf{m}} - \frac{\hat{\mathbf{m}}}{z}}{\Delta t} = \frac{1}{2} (\partial_x^2 - k^2) \left(\hat{\mathbf{m}} + \frac{\hat{\mathbf{m}}}{z} \right)$$

After rearranging and define $\bar{\mu} = k^2 + \rho R$ where $\rho = \frac{2(z-1)}{\Delta t(z+1)}$, the transformed momentum equations (in components) are:

$$(\partial_x^2 - \bar{\mu}^2) \hat{m}_1 = 0 \quad (4.50a)$$

$$(\partial_x^2 - \bar{\mu}^2) \hat{m}_2 = 0 \quad (4.50b)$$

Solving these 2 ordinary differential equations we obtain an expression for \hat{m}_1 and \hat{m}_2 :

$$\hat{m}_1 = A_1 e^{-\bar{\mu}x} \quad (4.51a)$$

$$\hat{m}_2 = A_2 e^{-\bar{\mu}x} \quad (4.51b)$$

where A_1 and A_2 are constants to be determined.

Note that unlike the intermediate velocities in Projection methods, the Gauge variable actually does not contain any spurious mode! This would indeed lead to an improved accuracy since there is no numerical boundary in \mathbf{m} .

This in turn guarantees the auxiliary field is also free of spurious modes. It is solved by the compatibility condition equation 4.49 e:

$$(\partial_x^2 - k^2) \hat{\chi} = \partial_x \hat{m}_1 + ik \hat{m}_2 \quad (4.52)$$

which then leads to an expression of $\hat{\chi}$ as:

$$\hat{\chi} = Pe^{-|k|x} + \frac{(-\bar{\mu}A_1 + ikA_2)}{\rho} e^{-\bar{\mu}x} \quad (4.53)$$

The velocities and pressure can then be recovered using the compatibility condition and the Gauge update formula respectively.

$$\begin{aligned} \hat{u} &= \hat{m}_1 - \partial_x \hat{\chi} \\ &= |k|Pe^{-|k|x} + \frac{(-k^2 A_1 + ik A_2 \bar{\mu})}{\rho} e^{-\bar{\mu}x} \end{aligned} \quad (4.54a)$$

$$\begin{aligned} \hat{v} &= \hat{m}_2 - ik \hat{\chi} \\ &= -ikPe^{-|k|x} + \frac{(A_2 \bar{\mu}^2 + ik A_1 \bar{\mu})}{\rho} e^{-\bar{\mu}x} \end{aligned} \quad (4.54b)$$

The pressure can be recovered using the transformed Pressure update formula:

$$\frac{\hat{p}}{\sqrt{z}} = \frac{1}{\Delta t} \left(1 - \frac{1}{z} \right) \hat{\chi} - \frac{1}{2} (\partial_x^2 - k^2) \left(1 + \frac{1}{z} \right) \hat{\chi}$$

Leading to

$$\hat{p} = \frac{z+1}{2 z^{1/2}} (\rho P e^{-|k|x}) \quad (4.55)$$

Unlike in projection method, the spurious mode must be eliminated with special pressure update formulas, the velocities and pressure in Gauge method naturally leads to the exclusion of spurious modes. Hence this demonstrates that no numerical boundary layers are formed in Gauge method. This is also confirmed by our numerical studies presented in Chapter 6.

Equipped with the boundary conditions given, we can then solve for the undetermined coefficients in the solutions above.

Normal component:

$$\hat{m}_1|_{x=0} = \hat{u}|_{x=0} = \hat{\alpha} \quad (4.56)$$

The tangential boundary condition for $\hat{u}|_{x=0}$ needs more attention, because by compatibility condition, solving \mathbf{m}^{n+1} requires the knowledge of χ^{n+1} which we don't have access yet:

$$\hat{u}|_{x=0} = \hat{m}_2|_{x=0} - ik\hat{\chi}^{n+1}|_{x=0} = \hat{\beta} \quad \text{exact relation} \quad (4.57)$$

An approximation to $\hat{\chi}^{n+1}$ is needed and similar to the projection methods, we use the same function: \hat{B} to represent the different approximations.

$$\hat{m}_2|_{x=0} = \hat{u}|_{x=0} + ik\hat{B}\hat{\chi}^{n+1}|_{x=0} \quad \text{actual boundary condition used in practice} \quad (4.58)$$

Combine equation 4.57 and 4.58 we get:

$$\hat{u}|_{x=0} + ik\hat{\chi}(\hat{B} - 1)|_{x=0} = \hat{\beta} \quad (4.59)$$

Same problem in regarding to the choice of auxiliary field approximation function \hat{B} . Hence it is worth to note that the tangential velocity field now does not satisfy the right boundary condition exactly (but to an order of accuracy depending on the choice of ϕ^{n+1} approximation). Hence we now need to ensure it approaches β along the boundary at a second order rate. Hence second order approximation to ϕ^{n+1} is needed and this corresponds to $\hat{B} = \frac{2z-1}{z^2}$ (For Alg 3 first order approximation to ϕ^{n+1} is sufficient since the approximation term is a multiplied with Δt by the compatibility equation).

Hence along the boundary (taking $x = 0$) we note the Gauge variables must satisfy:

$$A_1 = \hat{\alpha}, \quad m_1 \text{ at } x = 0 \quad (4.60a)$$

$$\hat{\beta} = A_2 - ik \left(\frac{2z-1}{z^2} \right) \left(P + \frac{(-\bar{\mu}A_1 + ikA_2)}{\rho} \right) \quad (4.60b)$$

Further because the auxiliary field satisfies a zero Neumann normal boundary condition ($\partial_x \hat{\chi} = 0$) hence we have:

$$\bar{\mu}^2 A_1 - ik\bar{\mu}A_2 - |k|\rho P = 0 \quad (4.61)$$

Summaries these conditions we can solve for A_1 , A_2 and P directly:

$$\begin{bmatrix} \bar{\mu}^2/R & -ik\bar{\mu}/R & -|k|\rho \\ ik\bar{\mu}\hat{B}(z)/R & (\rho + k^2\hat{B}(z)/R) & -ik\hat{B}(z)\rho \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ P \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{\beta}\rho \\ \hat{\alpha} \end{bmatrix} \quad (4.62)$$

where $\hat{B}(z) = \frac{2z-1}{z^2}$. Further it can be shown by a Taylor expansion argument that $\frac{1}{\hat{B}(z)} = \frac{z^2}{2z-1} = 1 + \mathcal{O}(\Delta t^2)$. Substitute this back into the linear system above (Equation 5.78) and then we can solve for A_1 , A_2 and P easily.

$$A_1 = \hat{\alpha} \quad (4.63a)$$

$$A_2 = \frac{\bar{\mu}^2\hat{\alpha} - (\bar{\mu} + |k|)(\bar{\mu}\hat{\alpha} - ik\hat{\beta})}{ik\bar{\mu}} + \mathcal{O}(\Delta t^2) \quad (4.63b)$$

$$P = \frac{(\bar{\mu} + |k|)(\bar{\mu}\hat{\alpha} - ik\hat{\beta})}{\rho|k|} + \mathcal{O}(\Delta t^2) \quad (4.63c)$$

The numerical solutions can then be recovered using Equation 5.70 and 5.71 by directly substituting A_1 , A_2 and P into the expressions for \hat{u} , \hat{v} and \hat{p} .

$$\left\{ \begin{array}{l} \hat{u}_{nu} = \left(\frac{(\bar{\mu} + |k|)(-|k|\hat{\alpha} + ik\hat{\beta})}{R\rho} + \mathcal{O}(\Delta t^2) \right) e^{-\bar{\mu}x} + \left(\frac{(\bar{\mu} + |k|)(\bar{\mu}\hat{\alpha} - ik\hat{\beta})}{R\rho} + \mathcal{O}(\Delta t^2) \right) e^{-\bar{\mu}x} \\ \hat{v}_{nu} = \left(\frac{i\bar{\mu}(\bar{\mu} + |k|) \left(-\frac{k}{|k|}\hat{\alpha} + i\hat{\beta} \right)}{R\rho} + \mathcal{O}(\Delta t^2) \right) e^{-\bar{\mu}x} + \left(\frac{-ik(\bar{\mu} + |k|)(\bar{\mu}\hat{\alpha} - ik\hat{\beta})}{R\rho|k|} + \mathcal{O}(\Delta t^2) \right) e^{-\bar{\mu}x} \\ \hat{p}_{nu} = \frac{z+1}{2z^{1/2}} \left(\frac{(\bar{\mu} + |k|)(\bar{\mu}\hat{\alpha} - ik\hat{\beta})}{R|k|} + \mathcal{O}(\Delta t^2) \right) e^{-\bar{\mu}x} \end{array} \right. \quad (4.64)$$

where nu denotes “numerical” to distinguish from the analytical solutions in Equation 5.9.

Now recall from the subsection Projection methods, we have shown that $\rho = s + \mathcal{O}(\Delta t^2)$, $\bar{\mu} = \mu + \mathcal{O}(\Delta t^2)$ and $\frac{z+1}{2z^{1/2}} = 1 + \mathcal{O}(\Delta t^2)$. Hence it is straightforward to show that the coefficients in the numerical solutions above are of second

order accuracy to that of the analytical solutions presented in Equation 4.9.

Hence with the Gauge variable tangential boundary equals to $\tau \cdot (\mathbf{u}^{n+1} + (2\chi^n - \chi^{n-1}))$, the Gauge method shows completely second order accuracy in both velocities and pressure. Also because neither the Gauge variable and the auxiliary field contains spurious modes, hence there is no special formula needed to filter out the numerical boundary layers as in Projection methods. This means the error convergence does not depend on the domain and boundary conditions as much as the projection methods which has fully second order accuracy only in periodic domains. The Gauge method exhibits the same order of accuracy in general domains too (see Shen' paper [14, 24] for details of proof).

Chapter 5

Model implementation

This chapter describes how the Navier Stokes numerical solver is actually constructed. We will talk about the spatial discretisation and implementation in Python. For clarity we will illustrate the numerical steps based on the Projection method with a lagged pressure approximation ($Pm\ 1(b)$). Some of the ideas for instance the use of ghost nodes are based on Roberts report on Navier stokes numerical solvers. This was one of the first reports I have read in this project. Hence I think it is important to include it here. The code is also available upon request.

5.1 Finite difference scheme

This section is concerned with the implementation of second order Finite difference scheme.

Before we have only considered time discretisations, here we will describe how the spatial discretisation is done in order to fully illustrate our numerical solver.

The idea of spatial discretisation is to divide the domain which contains infinitely many “points” into finite set of discrete cells or volumes. Then numerical techniques such as Finite difference and Finite volume methods can be used to approximate the continuous operators (e.g. Differentiation and Integration) based on the discrete cells.

There are many ways in which spatial discretisation can be done. Finite difference is one of the most simplest methods. Because our concern is mainly on the

accuracy of different projection methods, hence Finite difference method wins over other methods because of its easy implementation yet providing robust computation efficiency. There are of course disadvantages associated with using Finite difference schemes, including inability to handle complex geometries. Other more sophisticated methods including Finite elements and Finite volume methods can handle complex geometries but requires more complicated implementations. For instance, Finite element methods requires the use of Weak formulation of Navier Stokes equations. We leave these drawbacks as potential improvements in future researches.

The idea of finite differencing comes from using known function values to calculate derivatives and integrals based on the set of discrete points. For instance if our domain is an unit square, then we can divide it into $n \times n$ many cells each with width $\Delta x = \Delta y = \frac{1}{n}$. Then the function values and its derivatives can be calculated on these points. The width or step size (Δx) are important in the outcome of the numerical approximation. Small step size or finer grids usually gives better approximations and smaller (spatial) error.

5.1.1 Approximation to derivatives

The use of finite differencing to approximate derivatives are commonly used in numerical studies and simulations. The formula is derived from a simple Taylor expansion of the function $f(x, y, t)$. Suppose we want to find the first x derivative of f at location (i, j) corresponding to $(x = i\Delta x, y = i\Delta y)$ using a second order centred differencing scheme. We use the nearby function values at $i + 1$ and $i - 1$ to do so. The index i, j ranges from 0 to n, m respectively. Please don't be confused with time iterations "n". Let's Taylor expand $f_{i+1,j}$ and $f_{i-1,j}$ at $f_{i,j}$

$$f_{i+1,j} = f_{i,j} + f'_{i,j}(\Delta x) + \frac{f''_{i,j}(\Delta x)^2}{2} + \mathcal{O}(\Delta x^3)$$

$$f_{i-1,j} = f_{i,j} - f'_{i,j}(\Delta x) + \frac{f''_{i,j}(\Delta x)^2}{2} + \mathcal{O}(\Delta x^3)$$

Using $f_{i+1,j} - f_{i-1,j}$ to eliminate the second derivative we obtain:

$$\frac{\partial f}{\partial x} = f'_{i,j} = \frac{f_{i+1,j} - f_{i-1,j}}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

This is called "centered finite differencing" or 2 - point stencil (since 2 points are involved in calculations) and it is clearly of second order accurate. The derivative

in y and time uses the same formula with a change of indexing.

The second order and higher derivatives can be derived using the same Taylor expansion argument. For instance

$$\frac{\partial^2 f}{\partial x^2} = f''_{i,j} = \frac{f_{i+1,j} - f_{i,j} + f_{i-1,j}}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$

It is quite straightforward to generalise this Finite differencing schemes to higher order accuracy. However more points are involved as go up to higher order in Taylor series. This introduces many practical problems including larger stencil which makes implementation harder. Then the idea of “Compact Finite difference” comes into play where the derivatives at a set of points are solved by a linear system. This guarantees high order of accuracy without working with too many points. A 3 point stencil is often used when approximating the first derivative to 4th order accuracy. However due to limited time, this was not implemented in our solver but leaves to future researches.

5.1.2 Staggered grid

To this end, we have been working with grids where the function values (and its derivatives) are stored at the same locations: vertices (or centre) of the discrete cells. This is called “non-staggered” mesh grid or “collocated” grid. This the most “intuitive way” to think about the spatial discretisation, however this also introduces many problems including odd even grid decoupling because the pressure gradient and divergence terms are non-elliptic at grid scale [4]. This is because the variables are not stored in optimal locations. First introduced by Harlow, Francis H, Welch and J Eddie, staggered grid solves this issue quite nicely [16]. **staggered grid advantages**

Now with a staggered grid geometry, the variables are separated to different locations to avoid the odd-even grid decoupling. The scalar variable Pressure values are stored in the cell centres whereas the velocity variables are located at the cell faces. A picture of the staggered grid extracted from Robert’s report is shown below:

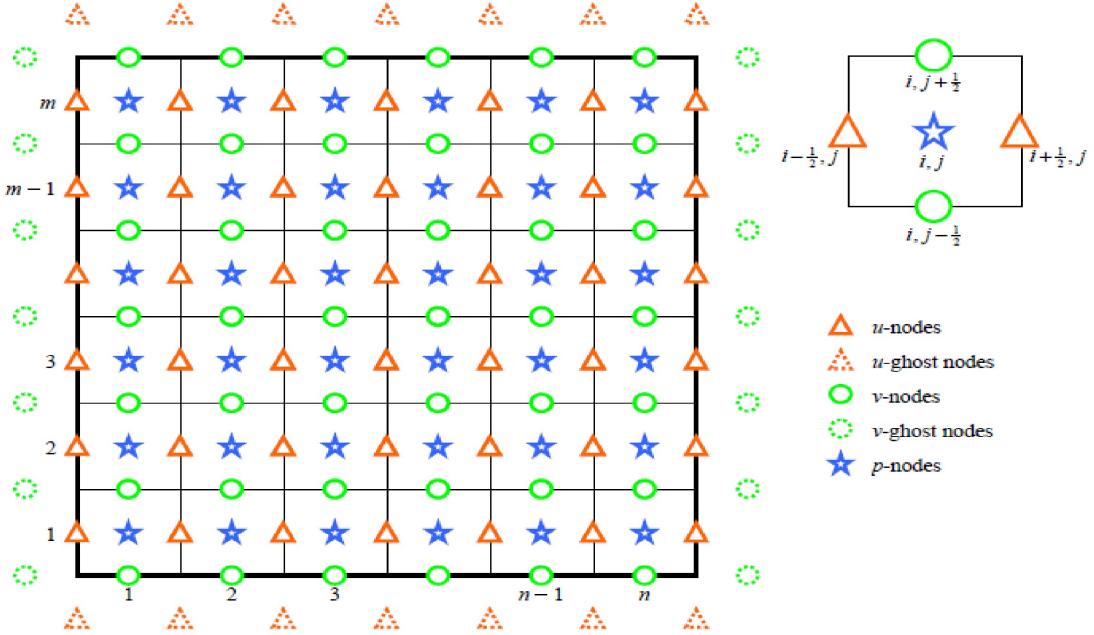


Figure 5.1: Picture of 2-D Staggered grid

The blue stars represents the pressure locations and corresponds to the indexing of $i = 1, 2, 3, \dots, n$ and $j = 1, 2, 3, \dots, m$; the red triangles are the “horizontal” u velocities have half indices and corresponds to $i - 1/2 = 1/2, 3/2, \dots, n + 1 - 1/2$ and $j = 1, 2, 3, \dots, m$; the “vertical” v velocities also have half indices and corresponds to $i = 1, 2, 3, \dots, n$ and $j - 1/2 = 1/2, 3/2, \dots, m + 1 - 1/2$. The boundary points are those stored at the edge of the domain. For instance, the u velocities have west and east boundary points with indices of West: $(i = 1/2, j)$ and East: $(i = n + 1/2, j)$; v velocities have North and South boundaries points with $(i, j = 1/2)$ and $(i, j = m + 1/2)$; whereas pressure only have interior points.

The derivatives can be calculated nicely in interior points however it is less straightforward to do so at the boundary points. For instance, if we want to know $\partial_y u_{i-1/2,1}$ next to the South boundary then this involves the up and down u values with index $i - 1/2, 0$ and $i = 3/2, 2$. However $u_{i-1/2,0}$ simply does not exists because it is outside our domain. Can we ignore it? obviously not because it is needed to calculate derivatives. The inability to specify boundary values is one of the major drawbacks of staggered mesh grids. To tackle this problem, many people have used the concept of “ghost cells” where we give u a value at location $(-1/2, j)$. Then the derivative operations can be conducted as normal as the interior points. Therefore interpolation is needed. Commonly,

people have used 2 point average to calculate $u_{-1/2,j}$. However this only gives first order accuracy and degrades the global spatial convergence error. To avoid this, higher order interpolations are needed. Robert has used a cubic polynomial interpolation to guarantee second order accuracy all the way up the boundary **citation of Roberts report**. The formula can be derived by using Taylor expansion of 3 interior points and the ghost cell at the boundary point. Then after rearrangement and cancelling of other function derivatives we obtain:

$$u_{i-1/2,0} = \frac{16}{5}u_{i-1/2,1/2} - 3u_{i-1/2,1} + u_{i-1/2,2} - \frac{1}{5}u_{i-1/2,3} \quad (5.1)$$

The value at the corner e.g. $u_{i-1/2,1/2}$ are boundary values which will be given by the Dirichlet boundary condition for u .

The North counterpart is calculated similarly:

$$u_{i-1/2,m+1} = \frac{16}{5}u_{i-1/2,m+1/2} - 3u_{i-1/2,m} + u_{i-1/2,m-1} - \frac{1}{5}u_{i-1/2,m-2} \quad (5.2)$$

Similarly the v velocities also suffer from this problem near the West and East boundaries and cubic interpolation is also used to calculate the ghost cells:

$$v_{0,j-1/2} = \frac{16}{5}u_{1/2,j-1/2} - 3u_{1,j-1/2} + u_{2,j-1/2} - \frac{1}{5}u_{3,j-1/2} \quad (5.3)$$

$$v_{n+1,j-1/2} = \frac{16}{5}u_{n+1/2,j-1/2} - 3u_{n,j-1/2} + u_{n-1,j-1/2} - \frac{1}{5}u_{n-2,j-1/2} \quad (5.4)$$

In Python, however we cannot work with “half indices”. Therefore a special treatment of index is needed. For instance, we can double the index values ($i \Rightarrow 2i$, $j \Rightarrow 2j$) so that half indices can be used. In our case, for the sake of simplicity, a re-indexing was used. Looking at the staggered grid again, by joining the triangles together (including ghost cells) we extract a grid for u velocities. Then Python index can be used here. For instance, West corresponds to $(i = 1/2, j)$ in staggered grid now have index $(i = 0, j)$ in our new grid. Same strategy is used for v and pressure too. It is worth to point out that the North and South are flipped between the staggered grid and Python grid.

5.1.3 Discretisation of Convective and Diffusive terms

Because for centered differencing, we want to calculate all of our variables (u , v and p) at half time indices. Therefore a temporal discretisation of convective and diffusive terms are also needed.

An explicit second order Adam Bashforth scheme is used to discretise the Convective term $[(\mathbf{u} \cdot \nabla) \mathbf{u}]^{k+1/2}$. Here k is used as time iterations.

$$[(\mathbf{u} \cdot \nabla) \mathbf{u}]^{k+1/2} = \frac{3}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^k - \frac{1}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{k-1} \quad (5.5)$$

The spatial discretisation follows from our first subsection with a change of indices to fit with the staggered mesh grid.

For u velocities which live on the triangles

$$\begin{aligned} [(\mathbf{u} \cdot \nabla) \mathbf{u}]_{i-1/2,j} &= u_{i-1/2,j} \frac{\partial u_{i-1/2,j}}{\partial x} \\ &+ v_{i-1/2,j} \frac{\partial u_{i-1/2,j}}{\partial y} = u_{i-1/2,j} \frac{u_{i+1/2,j} - u_{i-3/2,j}}{2\Delta x} \\ &+ v_{i-1/2,j} \frac{u_{i-1/2,j+1} - u_{i-1/2,j-1}}{2\Delta x} \end{aligned} \quad (5.6a)$$

However because in staggered grid, v and u are stored at different locations and hence spatial interpolation is again needed to compute $v_{i-1/2,j}$. This time a 4 point average will suffice for second order accuracy.

$$v_{i-1/2,j} = \frac{v_{i,j+1/2} + v_{i-1,j+1/2} + v_{i,j-1/2} + v_{i-1,j-1/2}}{4}$$

Similarly for the v component:

$$\begin{aligned} \mathbf{v} \cdot \nabla) \mathbf{v}]_{i,j-1/2} &= u_{i,j-1/2} \frac{\partial v_{i,j-1/2}}{\partial x} \\ &+ v_{i-1/2,j} \frac{\partial v_{i,j-1/2}}{\partial y} \\ &= u_{i,j-1/2} \frac{v_{i+1,j-1/2} - v_{i-1,j-1/2}}{2\Delta x} \\ &+ v_{i,j-1/2} \frac{v_{i,j+1/2} - v_{i,j-3/2}}{2\Delta x} \end{aligned} \quad (5.6b)$$

Once again 4 point average is used to compute $u_{i,j-1/2}$ which we do not have access to first.

$$v_{i,j-1/2} = \frac{v_{i-1/2,j} + v_{i+1,j} + v_{i+1,j-1} + v_{i-1,j-1}}{4}$$

There are other ways to treat the linear convective terms. We can put it into conservation form and use upwind scheme.

Crank Nicholson is used to treat the diffusive terms.

$$\nabla^2 \mathbf{u}^{k+1/2} = \nabla^2 (\mathbf{u}^* + \mathbf{u}^k) \quad (5.7)$$

where $\nabla^2 \mathbf{u}$ is discretised as

$$\nabla^2 u = \frac{u_{i+1/2,j} - 2u_{i-1/2,j} + u_{i-3/2,j}}{\Delta x^2} + \frac{u_{i-1/2,j+1} - 2u_{i-1/2,j} + u_{i-1/2,j-1}}{\Delta y^2} \quad (5.8a)$$

$$\nabla^2 v = \frac{v_{i+1,j-1/2} - 2v_{i,j-1/2} + v_{i-1,j-1/2}}{\Delta x^2} + \frac{v_{i,j+1/2} - 2v_{i,j-1/2} + v_{i,j-3/2}}{\Delta y^2} \quad (5.8b)$$

the divergence of velocity is a scalar field and hence needs to match up with the locations of pressure cells. it is calculated as:

$$\nabla \cdot \mathbf{u}_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \quad (5.9)$$

The gradient of pressure on the other hand is a vector field and hence needs to match up with the locations of velocities (triangles and circles). Hence it is calculated as follows:

x component:

$$\partial_x p_{i-1/2,j} = \frac{p_{i,j} - p_{i-1,j}}{\Delta x} \quad (5.10a)$$

$$\partial_y p_{i,j-1/2} = \frac{p_{i,j} - p_{i,j-1}}{\Delta y} \quad (5.10b)$$

5.1.4 Linear system solvers

Now with every individual terms fully discretised and the introduction of intermediate velocity fields the momentum equation becomes:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\nabla p^{n-1/2} - \frac{3}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^k + \frac{1}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{k-1} + \frac{1}{2R} (\mathbf{u}^* + \mathbf{u}^n) + F^{n+1/2} \quad (5.11)$$

where $F^{n+1/2}$ represents the external forcing, e.g. gravity.

Because we are working with $Pm 1(b)$, hence we have taken the pressure approximation $q = p^{n-1/2}$.

Because of the implicit treat of diffusive terms, the intermediate velocity field must be solved as a linear system.

Further rearrange the momentum equation:

$$\left(I - \frac{\Delta t}{2R} \nabla^2 \right) \mathbf{u}^* = \mathbf{u}^n + \Delta t \left(-\nabla p^{n-1/2} - \frac{3}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^k + \frac{1}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{k-1} + \frac{1}{2R} (\mathbf{u}^* + \mathbf{u}^n) + F^{n+1/2} \right) \quad (5.12)$$

Now with the Laplacian fully discretised, the problem above is a linear system to solve:

$$A\mathbf{u}^* = \mathbf{b} \quad (5.13)$$

where $A = \left(I - \frac{\Delta t}{2R} \nabla^2 \right)$ and \mathbf{b} equals to the right hand side of the above equation.

We use Algebraic multi-grid method provided in Scipy to solve it.

Poisson pressure.

This is directly resulted from projection:

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \quad (5.14)$$

with boundary condition:

$$\mathbf{n} \cdot \nabla p = 0 \quad (5.15)$$

Normalisation approach.

This Poisson equation is also solved by AMG multi-grid method. However care must be taken as the above equation with the Neumann boundary condition gives unique solution of ϕ only up to the addition of a constant. Hence we must find a way to recover the exact solution. We basically needs to put an extra constraint into it.

We call this new modified approach as “Pressure normalisation”.

Although not discussed in the literature of Projection methods, we infer that a normalisation process is also implemented in the literature by researchers.

There are many choices of normalisation. One common one is forcing the ϕ to satisfies an integral constraint:

$$\int_{\Omega} \int_{\Omega} \phi \, dx dy = 0 \quad (5.16)$$

Is this right? However this approach involves calculating integrals and due to the limited timing we did not use it. This could be done in future researches.

Setting

$$\phi_{nu}^{n+1} = \phi_{ex}^{n+1} + a \quad (5.17)$$

where the subscripts “nu” and “ex” stands for numerical and exact solutions respectively. Because of non-uniqueness there is no “exact” ϕ solution, here exact refers to the one required by the projection method. Other ϕ solutions would cause the pressure to converge differently as we have seen before.

In practice ϕ_{nu}^{n+1} is first solved using the Poisson equation with the zero Neumann boundary condition

Because the Poisson equation with Neumann boundary condition does not guarantee uniqueness of ϕ , hence we need to think what other constraints it must satisfy in the projection methods. It turns out that ϕ satisfies the pressure update equation! Rearrange equation (11) we obtain an expression for the constant a :

$$a = \phi_{nu}^{n+1} - \frac{1}{Re} \nabla \cdot \mathbf{u}^* + (p^{n+1/2} - p^{n-1/2}) \quad (5.18)$$

This requires the knowledge of $p^{n+1/2}$ which of course we don't have when solving for ϕ^{n+1} ! However remember that a is just a constant added to ϕ (and pressure), hence if we know the pressure field at one point in domain then we can fully recover a . This point is arbitrary, but in practice, it is usually chosen along the boundary or center or any other points easily accessed. Therefore we need to impose a “partial” Dirichlet condition on pressure. It is partial because we only require the knowledge of point in space. In this section, we have chosen the point from the boundary.

$$a = \phi_{nu}^{n+1} (x, y)|_{(x,y) \in \partial\Omega} - \frac{1}{Re} \nabla \cdot \mathbf{u}^* (x, y)|_{(x,y) \in \partial\Omega} + (p^{n+1/2} - p^{n-1/2}) (x, y)|_{(x,y) \in \partial\Omega} \quad (5.19)$$

Practically in order to ensure consistency we take 2 to 4 points along each boundary or corner and taking their average.

For the forced flow example we are considering, the analytical solution ($p = \sin(t) \cos(\pi x) \sin(\pi y)$) indicates the North and South boundaries ($y = \pm 1$) are essentially zero for domain $[-1, 1]^2$. Therefore naturally one can consider inhomogeneous Dirichlet boundary condition for pressure variable. Then by picking one arbitrary point along the boundary (for instance $y = 1$ and x is arbitrary) our equation for c reads:

$$a = \phi_{nu}^{n+1} (x, 1) - \frac{1}{Re} \nabla \cdot \mathbf{u}^* (x, 1)$$

In the case of Staggered grids the spatial point $(x, 1)$ corresponds to an index of $i, j = \frac{1}{2}$ where i stands for the horizontal index and j the vertical index. Dropping the time index for convenience we obtain:

$$a = \phi_{i,1/2}^{nu} - \frac{1}{Re} \nabla \cdot \mathbf{u}_{i,1/2}^*$$

However in Staggered grids, the coordinate $(i, 1/2)$ is along the top edge of the grid and no pressure is stored at this point (pressure is stored at (i, j) locations). Therefore interpolation from nearby cells are needed. This introduces spatial interpolation error and hence care must be taken to avoid accumulation of spatial errors.

In this case, we use a 3rd order cubic interpolation for $\phi_{i,1/2}^{nu}$ from 3 nearby points: $\phi_{i,1}^{nu}$, $\phi_{i,2}^{nu}$ and $\phi_{i,3}^{nu}$. Cubic interpolation is needed to ensure the first order derivative of $\phi_{i,1/2}^{nu}$ is also second order accurate along the boundary.

We Taylor expand $\phi_{i,1}^{nu}$, $\phi_{i,2}^{nu}$ and $\phi_{i,3}^{nu}$ at location $(i, 1/2)$ up to order $\mathcal{O}(\Delta t^3)$ and we obtained an interpolation formula for $\phi_{i,1/2}^{nu}$ as

$$\phi_{i,1/2}^{nu} = \frac{15}{8} \phi_{i,1}^{nu} - \frac{5}{4} \phi_{i,2}^{nu} + \frac{3}{8} \phi_{i,3}^{nu} \quad (5.20)$$

Exactly the formula can be used for $\nabla \cdot \mathbf{u}^*$ since they are stored at the same locations as ϕ .

Then the exact ϕ^{n+1} can thus be obtained by subtracting c from ϕ_{nu}^{n+1} point-wisely. Thus we minus the same constant a on every point of ϕ^{nu} . Then the correct pressure can be recovered and we call it the “Normalised Pressure”. This in theory should fix the problem of error of error enlargement at finer grids. And this method indeed works as we shall shortly!

For Projection method *Pm 2*, exactly the process is used. In fact the expression for c is identical.

For Gauge method, exactly the same procedure can be implemented too. From solving the Poisson equation we know that:

$$\chi_{nu}^{n+1} = \chi_{ex}^{n+1} + b \quad \text{and} \quad \chi_n^n = \chi_{ex}^n + c \quad (5.21)$$

where b and c are constants. They are generally not equal and hence Gauge method also suffers from the non-uniqueness problem.

Again the point where we evaluate b and c is arbitrary, so let's use the same one considered above.

Substitute the expression of χ_{nu} into the pressure update equation and rearranging we obtain:

$$b - c = \chi_{i,1/2}^{n+1, nu} - \chi_{i,1/2}^{n, nu} + \frac{\Delta t}{2 Re} (\nabla \cdot m_{i,1/2}^{n+1} + \nabla \cdot m_{i,1/2}^n) \quad (5.22)$$

5.1.5 update velocity and pressure.

This is the easiest step:

The velocities are updated as follows:

By projection step:

new u velocity:

$$u_{i-1/2,j}^{k+1} = u_{i-1/2,j}^* - \Delta t \partial_x p_{i-1/2,j} = u_{i-1/2,j}^* - \Delta t \frac{p_{i,j} - p_{i-1,j}}{\Delta x} \quad (5.23a)$$

and new v velocity

$$v_{i,j-1/2}^{k+1} = u_{i,j-1/2}^* - \Delta t \partial_y p_{i,j-1/2} = u_{i,j-1/2}^* - \Delta t \frac{p_{i,j} - p_{i,j-1}}{\Delta y} \quad (5.23b)$$

and finally pressure:

$$p_{i,j}^{n+1/2} = p_{i,j}^{n-1/2} + \phi_{i,j}^{n+1} - \frac{\Delta t}{2R} \nabla^2 \phi_{i,j}^{n+1} = p_{i,j}^{n-1/2} + \phi_{i,j}^{n+1} - \frac{1}{2R} \nabla \cdot \mathbf{u}_{i,j}^* \quad (5.24)$$

Chapter 6

Numerical Results

In this chapter, we present the results generated by our numerical solvers based on projection methods: Alg 1, Alg 2 and Alg 3, as well as the Gauge method. We investigate each method's accuracy through numerical tests and visual simulations.

Numerical validation of Projection methods

6.1 Accuracy test

There are many ways to test the accuracy of numerical schemes. One of the most popular way is to compare the numerical solutions generated through computer simulations with analytical solutions. By calculating the error with appropriate norms (e.g. L_1 , L_2 and L_∞) this method provides precise measure on error and convergence rate. In case where analytical solutions are not known or difficult to implement, people often use bench mark results. These results are not the analytical solutions to the test problem, rather they are numerical solutions generated in fine grids that are widely accepted in literature. For instance, the lid driven cavity problem is such a simple test problem which shows turbulent flow patterns. We will give a qualitative illustration of the classic cavity problem in this chapter too.

For the main part of this chapter because we are aiming to conduct accuracy and convergence test of projection methods, hence we consider two simple 2-D flow problems where analytic solutions do exist. This reveals more fundamental properties and error behaviours for the projection schemes.

6.1.1 Forced flow

The test problem is a forced flow example where the fluid is confined in a square domain $\Omega = [-1, 1] \times [-1, 1]$ and initially remains stationary. The flow is initiated by an external driving force. This is one of the simplest settings to consider slip/no-slip boundary conditions. In this case an homogeneous boundary condition (no - slip) is imposed on the four sides of the domain
 $(\partial\Omega : \text{West: } x = -1, \text{ East: } x = 1, \text{ South: } y = -1, \text{ North: } y = 1)$.

The analytical solution satisfies:

$$\begin{cases} u = \pi \sin(t) \sin(2\pi y) \sin^2(\pi x) \text{ in } \Omega \\ v = -\pi \sin(t) \sin(2\pi x) \sin^2(\pi y) \text{ in } \Omega \\ p = \sin(t) \cos(\pi x) \sin(\pi y) \text{ in } \Omega \\ u_0 = u(t=0, x, y) \text{ in } \Omega \\ v_0 = v(t=0, x, y) \text{ in } \Omega \end{cases} \quad (6.1)$$

with homogeneous Dirichlet boundary condition:

$$u(t, x = \pm 1, y) = v(t, x, y = \pm 1) = 0$$

It is augmented with a forcing term: $f = \partial_t \mathbf{u} - \frac{1}{R} \nabla^2 \mathbf{u} + \nabla p$ to make the above “solutions” valid. For simplicity we take the Reynolds number (R) to be 1. By substituting our analytical solutions into f we find the forcing term is:

$$f = \begin{cases} fx = \pi \cos(t) \sin(2\pi y) \sin^2(\pi x) - 2\pi^3 \sin(t) \sin(2\pi y) (\cos(2\pi x) - 2\sin^2(\pi x)) \\ \quad - \pi \sin(t) \sin(\pi y) \sin(\pi x) \text{ in } \Omega \\ fy = -\pi \cos(t) \sin(2\pi x) \sin^2(\pi y) - 2\pi^3 \sin(t) \sin(2\pi x) (2\sin^2(\pi y) - \cos(2\pi y)) \\ \quad + \pi \sin(t) \cos(\pi x) \cos(\pi y) \text{ in } \Omega \end{cases} \quad (6.2)$$

The CFL number is given as: $CFL = U \Delta t / \Delta x$ where $U = \pi$ is the absolute bound for velocity. We want the CFL number to be below one to ensure stability. However too small CFL number will decrease the efficiency because the numerical solver would take a longer time to run to incorporate small time stepping. Hence we have chosen the optimal CFL number to be 0.5. This is also a commonly used value [6].

The actual error e_h is consisted of both spatial and temporal error and h is just an index to indicate the grid size (e.g. a 120×120 grid).

In our purpose of testing the temporal error behaviour, we want the temporal error to dominate and thus we can assume the following simplified error relation:

$$e_h = \Delta t^p \quad (6.3)$$

where p is the convergence rate or order of accuracy (don't be confused with pressure) and in our case we want $p = 2$.

The error e_h is simply calculated by taking the difference between numerical and analytical solutions point-wisely. It is important to compare the solutions only under the same conditions, for instance sane grid sizes and end of time... Comparing solutions at different time would of course cause strange error behaviours!

By taking the logarithm on both side of the equation above we see that the logarithm of error in time has a linear relationship with the logarithm of time stepping (Δt). Thus the slope of the log error function should be the convergence rate p .

$$\log_{10}(e_h) = p \log_{10}(\Delta t) \quad (6.4)$$

Base 10 is used for simplicity, other bases including 2 and natural exponential works the same.

The strategy is follows: with the error of a coarse grid denoted as e_h . We half the spatial stepping (Δx) or equivalently double the grid size (e.g. from 15×15 to 30×30). Denoting the error of the finer grid as $e_{h_{1/2}}$. The time stepping mean while is also halved. Then we look at how the error decreases. If our numerical scheme is second order accurate then the error should decrease by a factor of 4 going from the coarser grid to finer grid.

A Log-Log plot of the error function is then generated and the convergence rate is finally extracted as the slope of the log error function.

It is important to specify the error norm. As noted by many researchers that different norms could generated different convergence rates! [15, 24]

In our analysis 3 commonly used norms were implemented: L_1 , L_2 and L_∞ . Each could reveal certain aspect of error behaviour. By combing the results using these 3 norms, a comprehensive analysis of convergence rate can be obtained.

We take the normalised L_1 norm and it is defined as:

$$\| e^n \|_{L_1} = \frac{1}{M} \sum_{m=0}^M |e_m^n| \quad (6.5)$$

where e^n is the error field measured at time iteration n and M is the dimension or size of the error field. For instance a 15×15 grid corresponds to $M = 15^2$.

similarly we define the L_2 norm to be:

$$\| e^n \|_{L_2} = \left(\frac{1}{M^2} \sum_{m=0}^M |e_m^n|^2 \right)^{1/2} = \frac{1}{M} \left(\sum_{m=0}^M |e_m^n|^2 \right)^{1/2} \quad (6.6)$$

and L_∞ :

$$\| e^n \|_{L_\infty} = \max_{0 \leq m \leq M} |e_m^n| \quad (6.7)$$

To achieve optimal results, there are a number of things that need to be considered.

First, since we are aiming to investigate the temporal error and hence we need to use fine grids (small spatial stepping: $\Delta x \ll 1$) to ensure the spatial error is negligible. Formally we want: $\Delta x^2 \ll \frac{1}{R} \Delta t$. Hence we have run the solvers up to a grid size of 240×240 and also 480×480 sometimes. Higher grid numbers are also possible but takes very long time to compute, which would decrease the efficiency. For completeness we have also included the results for small to moderate grids. Overall we have maximum 6 data points corresponding to grid size of 15, 30, 60, 120, 240 and 480.

Second, care must also be taken to the pressure gradient along the boundary. We need to make sure the normal pressure gradient is non-zero along the boundary so that the accuracy performance can be distinguished between projection method Alg 1 and Alg 2. Fortunately, for our forced flow solutions, the normal pressure gradient is non-zero along the North and South boundaries ($y = \pm 1$) (whereas it is zero along the other two boundaries).

With the above conditions all satisfied, we can now finally test our numerical solutions! As for a start, we run the simulations at $t = 1$ with the spatial domain: $[-1, 1]^2$ and homogeneous Dirichlet boundary conditions imposed on velocities.

The exact and numerical solutions are summarised in Figure 6.1. The numerical solutions are computed by Alg 2 with grid size: 60×60 and $CFL = 0.5$. The

numerical solutions seem to do a good job in approximating the true solutions. These “nice” graphs however do not reveal much information about the error behaviours. Hence we need to look at the convergence rates instead.

We first compare Alg 1 and Alg 2 results to illustrate the improvement of modified pressure update formula (Equation 3.32 a) with $q = p^{n-1/2}$). The errors for grid size 30 to 120 are summarised in Table 1. In addition the convergence rates for both U and V velocities are highly identical, hence we only report the U component results in the subsequent analysis. It is clear that the velocities in both Alg 1 and Alg 2 show second order convergence rates across all norms whereas in pressure they do not. For Alg 1, L_1 and L_2 norms show first order convergence rate (1.29 and 1.21 respectively) whereas L_∞ shows an even deteriorated rate of 0.32. The sudden drop in convergence rate in L_∞ indicates the presence of numerical boundary layer, because L_∞ picks up the large errors or singularities better.

For Alg 2, the results in pressure are better, but still not very satisfied. L_1 norm shows a close to second order rate of 1.92 whereas L_2 shows a decreased rate of about 1.74. The L_∞ then further decreased to only about first order rate (1.14). The degradation of accuracy in L_∞ also indicate the presence of large singularities, whereas L_1 , L_2 tend to smooth them out. The overall magnitude of error is also relatively smaller in Alg 2. Take a look at the pressure error field in Figure 6.2, we found a pronounced numerical boundary layer in Alg 1 corresponding to $y = \pm 1$. Note that the normal gradient of pressure at these two boundaries is $\mathbf{n} \cdot \nabla p|_{y=\pm 1} = \mp\pi \sin(t) \cos(\pi x) \neq 0$. Hence this indicates clearly the numerical boundary layer is very much formed by the inconsistent normal pressure gradient resulted from the old pressure updated formula used in Alg 1. In Alg 2 the numerical boundary layer seems still exists but reduced to four thinner spikes located at the corners of the domain. In addition, the magnitude of the boundary layer in Alg 2 is also about 4 times smaller. All these observations indicate that the modified pressure update formula in Alg 2 do seem to filter out the spurious mode contained in the intermediate velocity field and auxiliary field better, and hence gives better pressure error convergence. This is consistent with our predication in normal mode analysis and also matches with the findings from other researchers like David, Strikwerda and Shen et al [6, 14, 15, 28].

Error in U velocity					
Alg 1	Error norm	30×30	60×60	120×120	rates
	L_1	3.75e-4	9.34e-5	2.32e-5	2.01
	L_2	4.66e-4	1.16e-4	1.88e-5	2.0
	L_∞	1.11e-3	2.77e-4	6.89e-5	2.02
Alg 2	L_1	3.23e-4	8.01e-5	1.99e-5	2.01
	L_2	4.09e-4	1.01e-4	2.53e-5	2.00
	L_∞	9.23e-4	2.33e-4	5.81e-5	2.00
Error in Pressure					
Alg 1	Error norm	30×30	60×60	120×120	rates
	L_1	6.96e-3	1.77e-3	7.29e-4	1.29
	L_2	1.02e-2	3.07e-3	1.33e-3	1.21
	L_∞	6.64e-2	3.87e-2	3.01e-3	0.32
Alg 2	L_1	1.26e-3	2.99e-4	8.32e-5	1.92
	L_2	1.68e-3	4.59e-4	1.47e-4	1.74
	L_∞	9.36e-3	6.31e-3	3.24e-3	1.14

Table 6.1: Error in U velocity and pressure for Alg 1 and Alg 2, $cfl = 0.5$

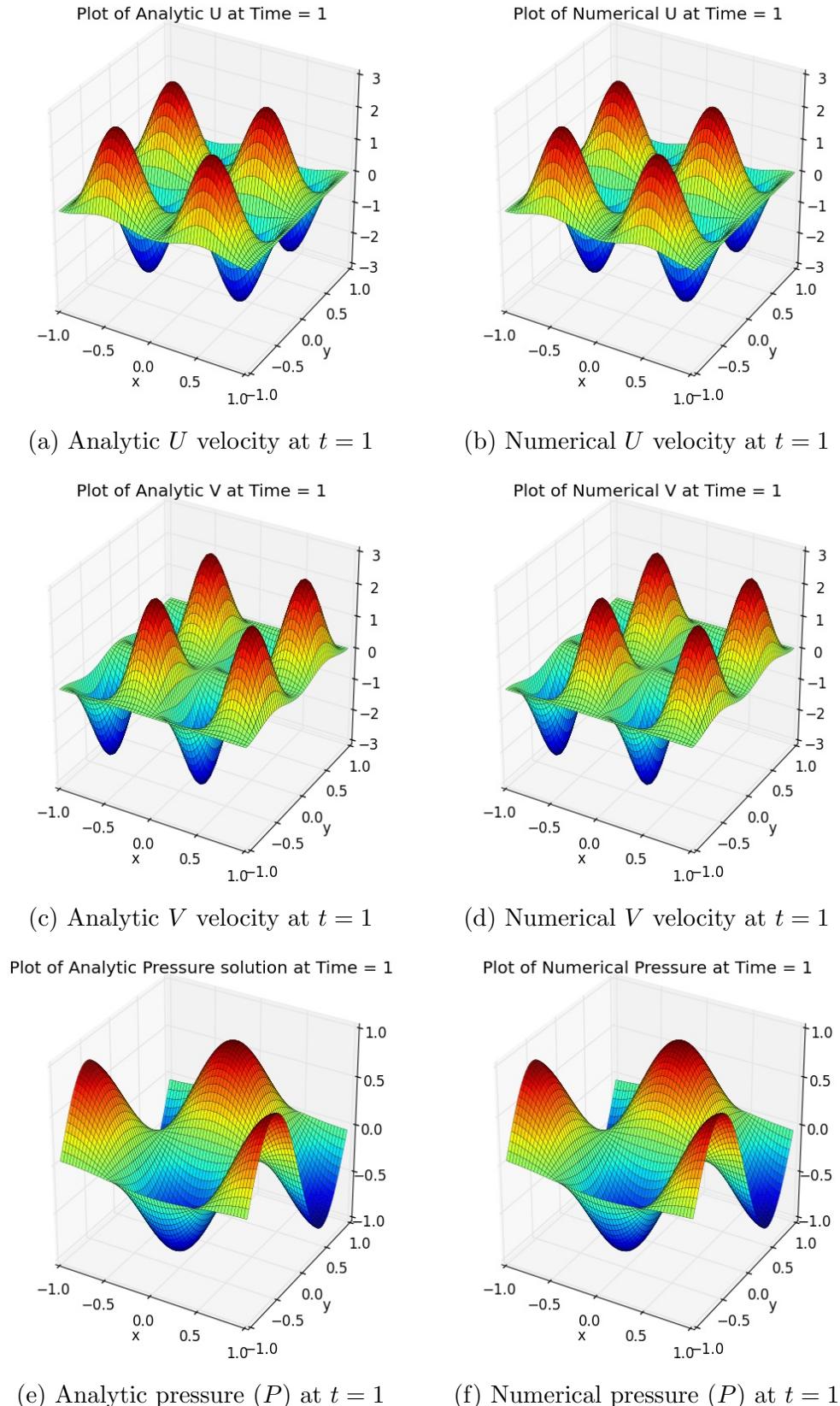


Figure 6.1: Plot of exact and numerical solutions (U, V, P) at time $t = 1$ on the spatial domain of $[-1, 1]^2$ with grid size 60×60 , Alg 2 was used to compute the numerical solutions with $CFL = 0.5$

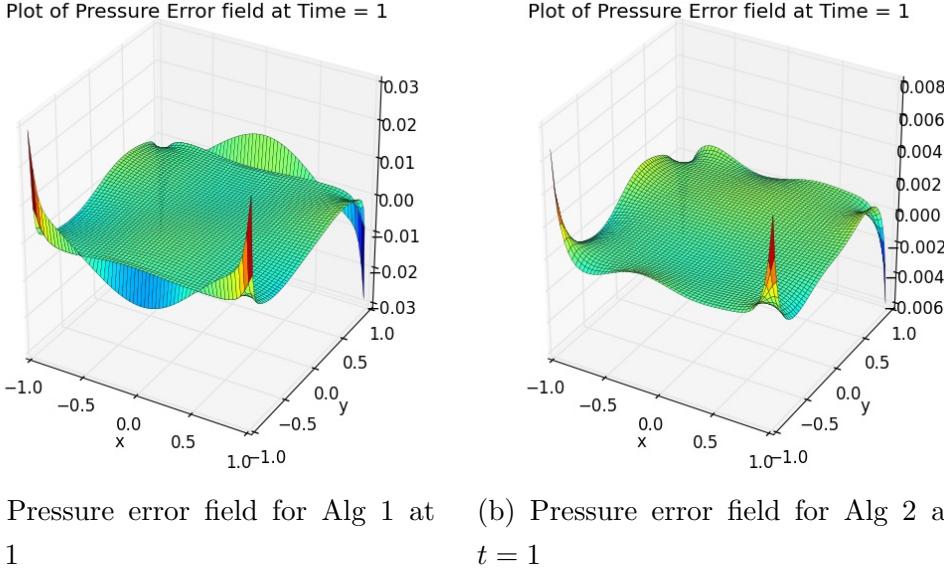


Figure 6.2: 3D surface plot of pressure error field for Alg 1 and Alg 2 at time $t = 1$ on the spatial domain of $[-1, 1]^2$ with grid size 60×60 and $CFL = 0.5$

We then compare with the schemes that were all predicated to be second order accurate, namely Alg 2, Alg 3 and Gauge method. The same test problem is used with the same domain ($[-1, 1]^2$) and end time to ensure consistency. This time only the convergence in pressure error and divergence of intermediate velocity fields are shown because the convergence in velocities in all methods are consistently second order. We focus more on the error in pressure.

The Log-Log plot for errors in pressure are summarised below. We have run the code from 15×15 up to the 480×480 the finest. Even finer grids are also able to be calculated by increasing the number of cells, but this is omitted due to the exceedingly long hours take in solving the Poisson equations.

Interestingly we observe an outlier at grid size 15 where the pressure error is very large compared to the those at the finer grids. This is observed in all methods for this test problem. This is most likely due to the large spatial error which dominates at such a small grid size. Hence for the sake of reliability, we excluded these data points when computing the convergence rates. The errors after grid 60 reduces consistently and converge asymptotically to the second order reference slope (especially for Alg 3 and Gauge method). This asymptotic convergence behaviour is due to the gradual phasing out of spatial errors dominance at finer

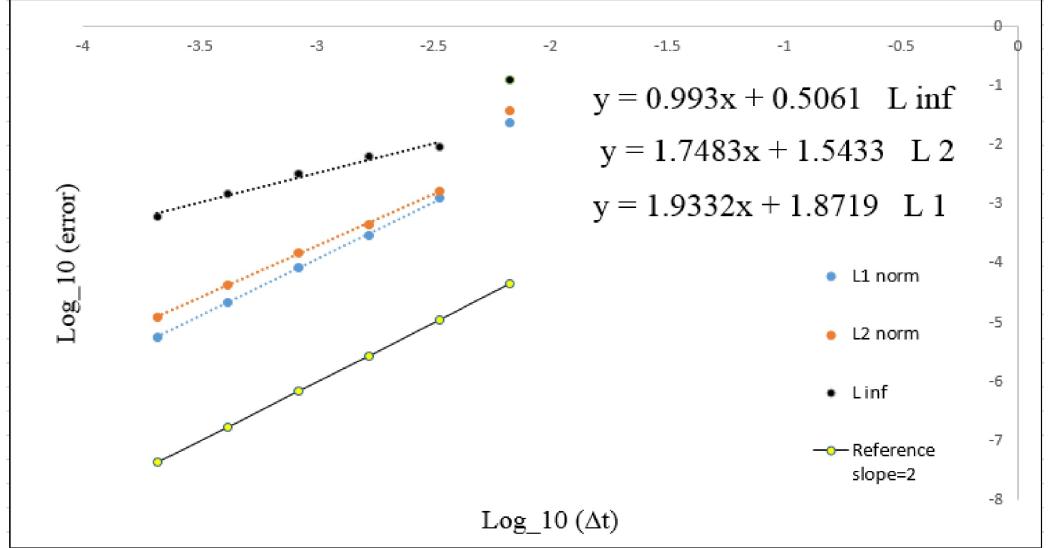
and finer grids. This is the most common error behaviour in Projection methods and our results now fully line up with those found in literature [15].

Even though Alg 3 and Gauge methods show optimal 2nd order convergence for this test problem, the projection method Alg 1 however shows deprecated convergence especially measured in L_∞ . This is because the four spikes in the pressure error plot for Alg 2 before. The pressure error field for Alg 3 and Gauge method shows smooth boundary layers instead. Interestingly the convergence rates in $\nabla \cdot \mathbf{u}^*$ is higher in Alg 2 than Alg 3 with about 1.5 to 1.8 order accuracy compared to 0.4 to 0.8 respectively. This is consistent with our hypothesis that the intermediate velocity field approximates the true velocity fields better in Alg 2.

Taking a look at the surface plot of $\nabla \cdot \mathbf{u}^*$ in Figure 6.6 both methods shows the presence of numerical boundary layers because we know that $\nabla \cdot \mathbf{u}^*$ contains the spurious mode ($e^{-\gamma x}$). Hence, our results indicate that the numerical boundary layer contained in the auxiliary variables in projection methods are better filtered out in Alg 3 than Alg 2. The exact cause of the deterioration in performance in Alg 2 still remains puzzling. Because both methods use the same pressure update formula, hence we infer that it is the boundary condition for \mathbf{u}^* and pressure approximation which affects the performance. As discussed in the normal analysis chapter that we know that Alg 2 has an inconsistent normal pressure gradient resulted from the choice of pressure approximation q . When $q = p^{n-1/2}$ as in Alg 2 this is “forcing” the normal pressure gradient at time $(n + 1/2)\Delta t$ to be zero which is inconsistent with that of the analytic counterpart at boundary $y = \pm 1$. Then numerical boundary layer cannot be completely filtered out. However this explanation is not very satisfactory as the normal analytic pressure gradient is zero independent of time at boundaries $x = \pm 1$, but we clearly observes four spikes in each corner of the domain the pressure error plot. We will rather illustrate the inconsistent choice of q in a different test problem. The problem is possibly not caused by boundary conditions choice of \mathbf{u}^* because we see a good convergence rate in $\nabla \cdot \mathbf{u}^*$.

Taking a closer look at the the plot of the analytic pressure field, it seems like the pressure values rapidly to zero approaching the North and South boundaries at $y = \pm 1$. This implies large change in pressure gradients along boundary. We infer that the cubic spatial interpolation of “ghost cells” implemented in a

staggered grid solver causes the pressure gradients to have a sharp increase along boundary. Then combined with the inconsistent choice of q we see a degradation of accuracy in pressure in Alg 2. As suggested by Brown (although a different test problem), the large numerical boundary layer problem in $\nabla \cdot \mathbf{u}^*$ could be reduced by making velocity variables to satisfy a “free boundary condition” where its third order derivative is zero. Then this might help to increase the convergence rate in pressure [6]. However this is problem dependent and due to limited space and time we did not implement this.



(a) Log-Log plot of Convergence rate for Pressure

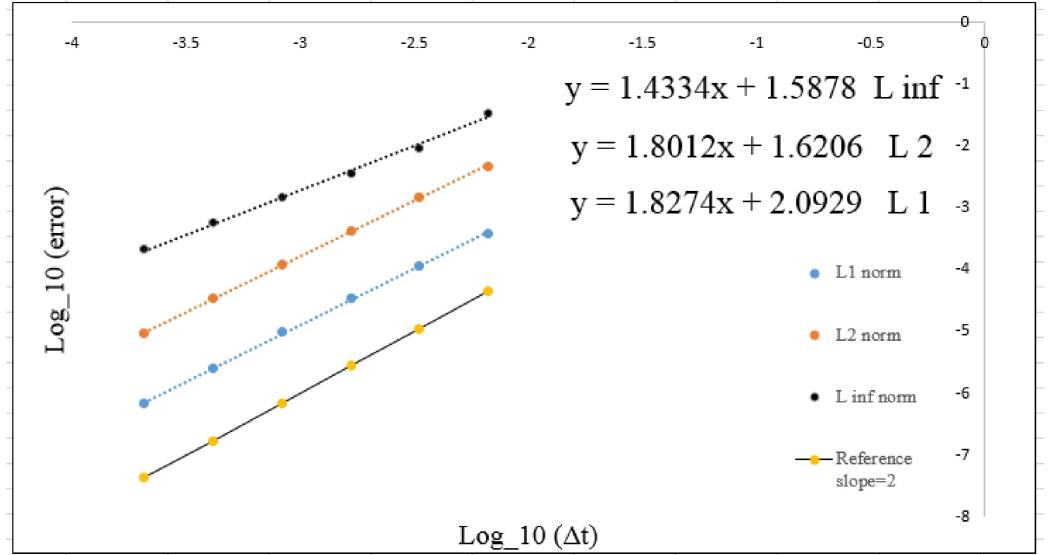
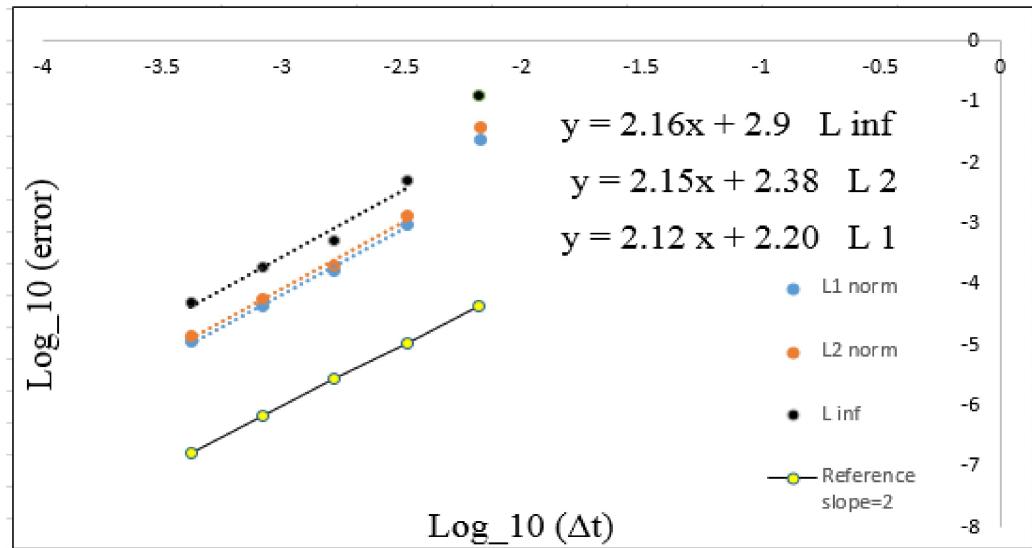
(b) Log-Log plot of Convergence rate for $\nabla \cdot \mathbf{u}^*$.

Figure 6.3: Plot of Convergence rates for Alg 2 with Normalised Pressure approach used. Domain: $[-1, 1]^2$, time = 1 and CFL = 0.5. In each plot, the data points corresponding to grid sizes of 15, 30, 60, 120, 240, and 480. Subplot (b) shows the error in the divergence of intermediate velocity. The data points between different norms are close to each. Hence L_1 and L_∞ norms were shifted up and down by 0.5 respectively to distinguish from L_2 norm data points.



(a) Log-Log plot of Convergence rate for Pressure

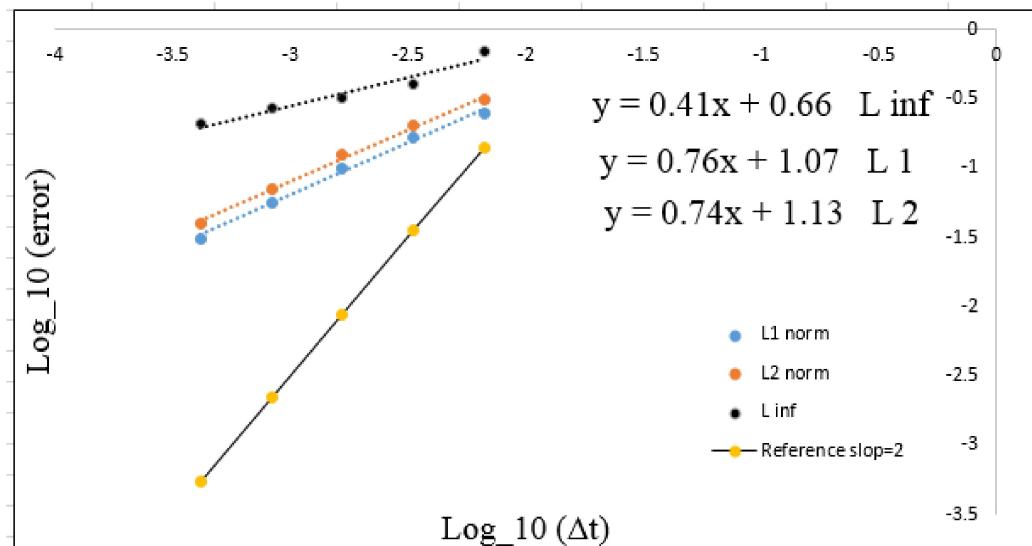
(b) Log-Log plot of Convergence rate for $\nabla \cdot \mathbf{u}^*$.

Figure 6.4: Plot of Convergence rates for Alg 3 with Normalised Pressure approach used. Domain: $[-1, 1]^2$, time = 1 and CFL = 0.5. In each plot, the data points corresponding to grid sizes of 15, 30, 60, 120, 240.

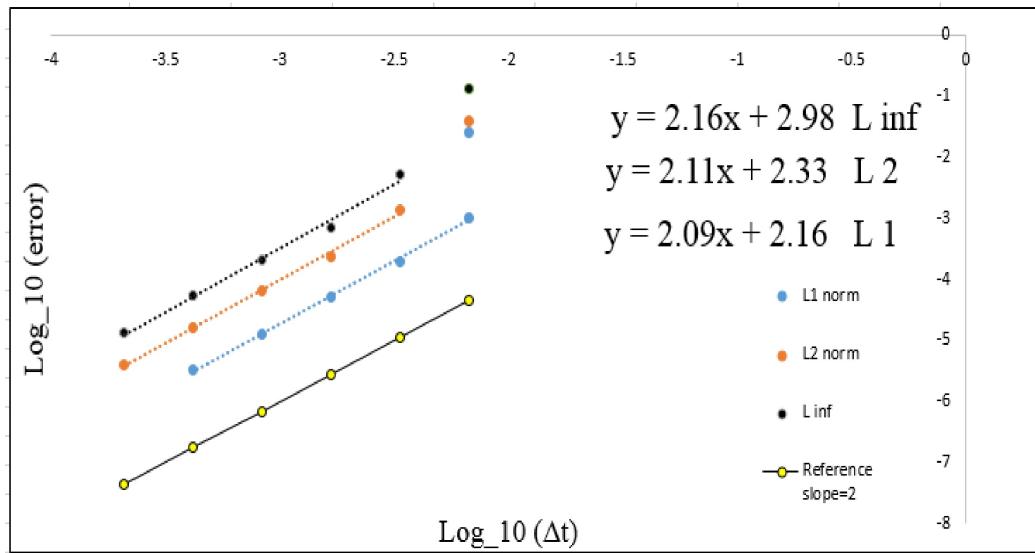
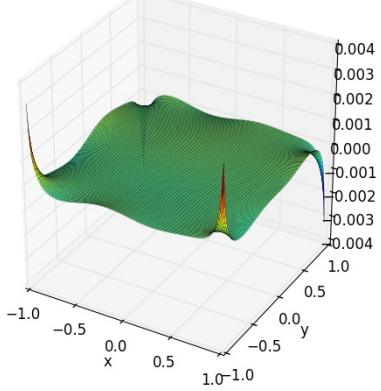
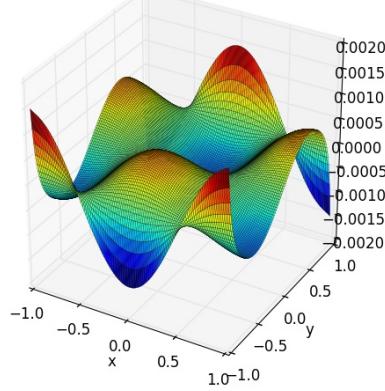


Figure 6.5: Long-Log plot of Pressure Convergence rates for Gauge method with Normalised Pressure approach used. Domain: $[-1, 1]^2$, time = 1 and CFL = 0.5. The data points corresponding to grid sizes of 15, 30, 60, 120, 240, and 480.

Plot of Pressure Error field at Time = 1

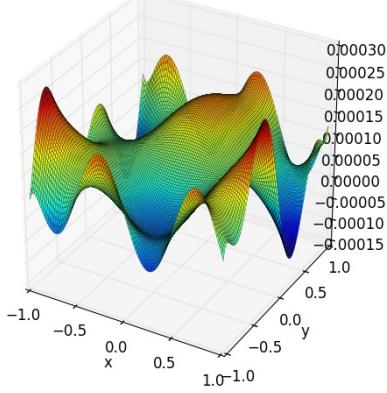


(a) Pressure error field for Alg 2 method

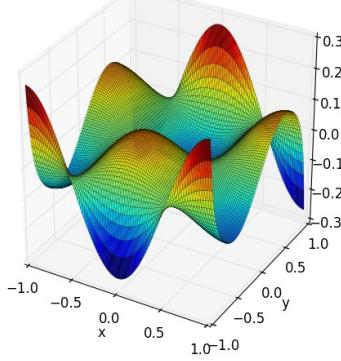
Plot of Divergence of intermediate velocity u^* at Time = 1

(b) Divergence of intermediate velocity field for Alg 2 method.

Plot of Pressure Error field at Time = 1

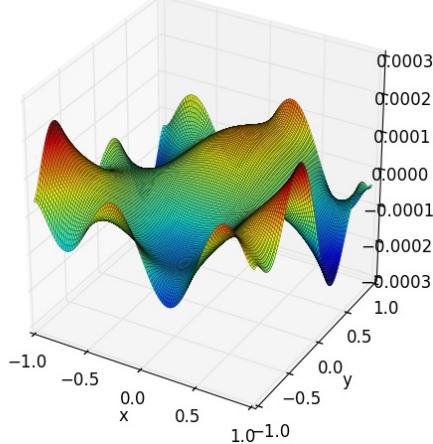


(c) Pressure error field for Alg 3 method

Plot of Divergence of intermediate velocity u^* at Time = 1

(d) Divergence of intermediate velocity field for Alg 3 method.

Plot of Pressure Error field at Time = 1



(e) Pressure error field for Gauge method.

Figure 6.6: Plots of Pressure error fields and divergence of intermediate velocity for Alg 2, Alg 3 and Gauge method at grid size of 120. Domain: $[-1, 1]^2$, time = 1 and CFL = 0.5.

Periodic boundary

To investigate the performance of projection methods in different boundary conditions, we have implemented a periodic boundary condition in y in order to mimic a periodic channel geometry. The same forced flow problem is considered, but in domain $[-\frac{1}{2}, \frac{1}{2}]^2$. Periodic boundary is ensured by putting the “ghost cell” values at one side of domain to be equal to the interior points just next to the opposite boundary. For instance, for U velocity at the South boundary ($y = -\frac{1}{2}$) we impose the condition: $u_{i-1/2,0} = u_{i-1/2,m}$. This eliminates the spatial interpolation error occurred in calculating the “ghost cell” values. The exact solutions are once again illustrated below in Figure 6.7. The convergence rates in pressure are also summarised.

Interestingly, this time we see all the methods including Alg 2 show second order error convergence in pressure! The L_∞ in Alg 2 is improved to 1.90 (as compared to 1.01 in the previous domain). The pressure error fields and the plot of divergence of intermediate velocity fields for projection methods are also more smooth now. In fact the pressure error fields for Alg 2, Alg 3 and the Gauge method is very similar now, indicating that the periodic geometry recovers second order accuracy for projection methods. This is in agreement with our normal mode analysis done before in Chapter 4 and this also indicate the projection methods (in particular Alg 2) only show optimal pressure error convergence in special domains like the periodic boundary domain whereas the Gauge method shows optimal convergence in general domains too. This finding is in support of Shen et.al results [14, 15] where they have also given an error bound of 1.5 for projection method (Alg 2 in particular). This shows a limitation in normal mode analysis too. The exact cause of this problem however still remains open in literature.

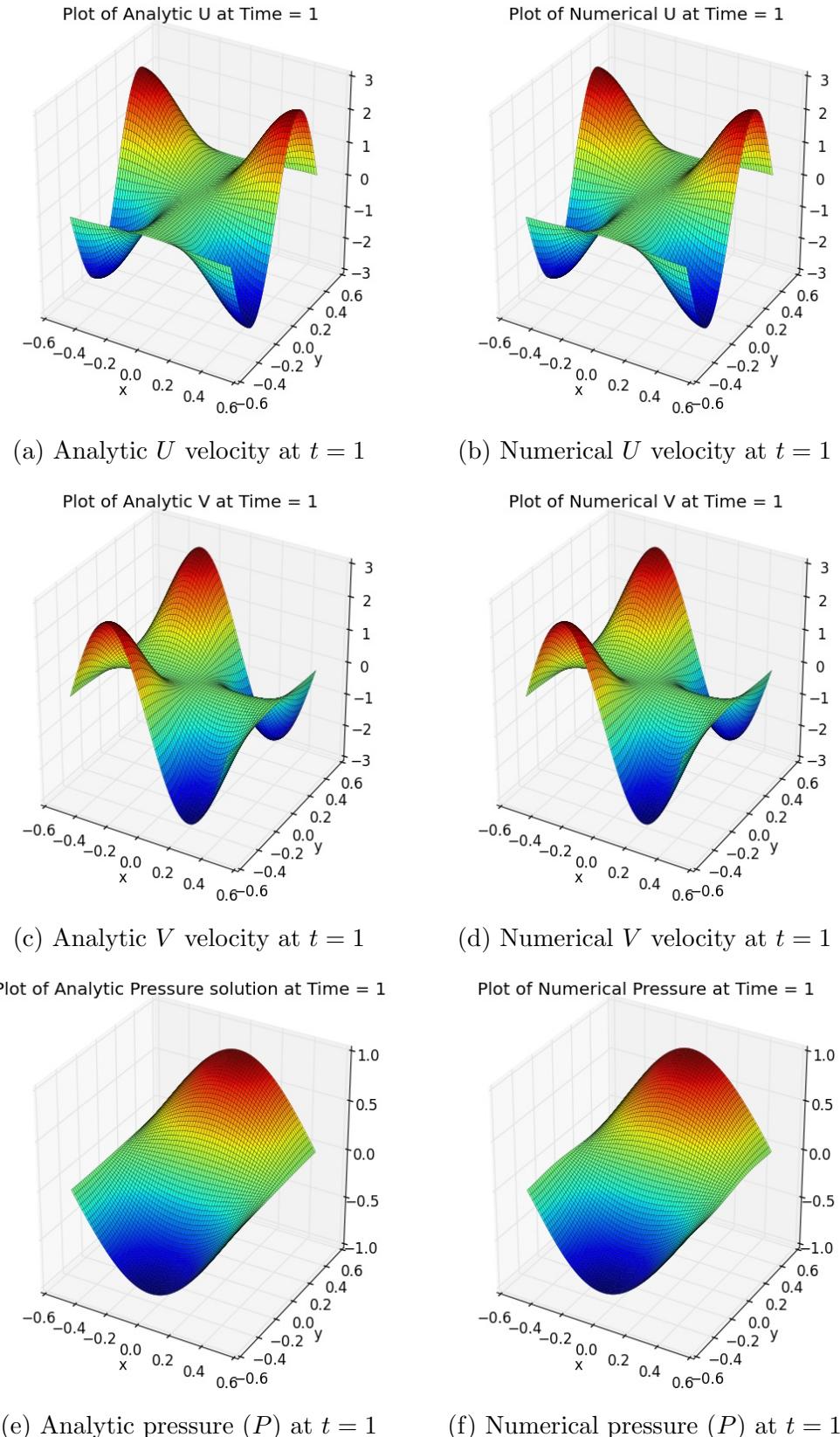
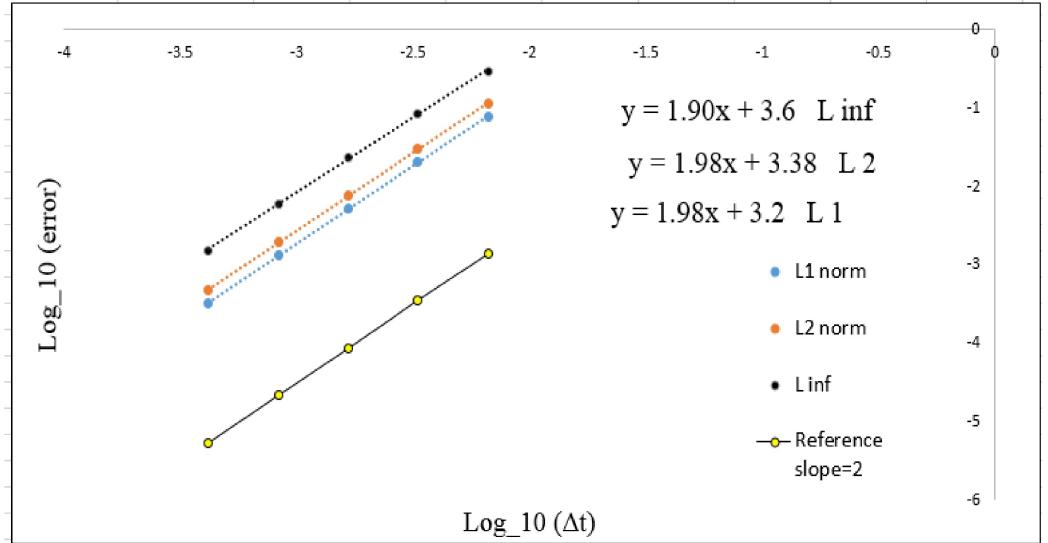


Figure 6.7: Plot of exact and numerical solutions (U, V, P) at time $t = 1$ on the spatial domain of $[-\frac{1}{2}, \frac{1}{2}]^2$ with grid size 60×60 . Alg 2 method is used to compute the numerical solutions with $CFL = 0.5$



(a) Log-Log plot of Convergence rate for Pressure

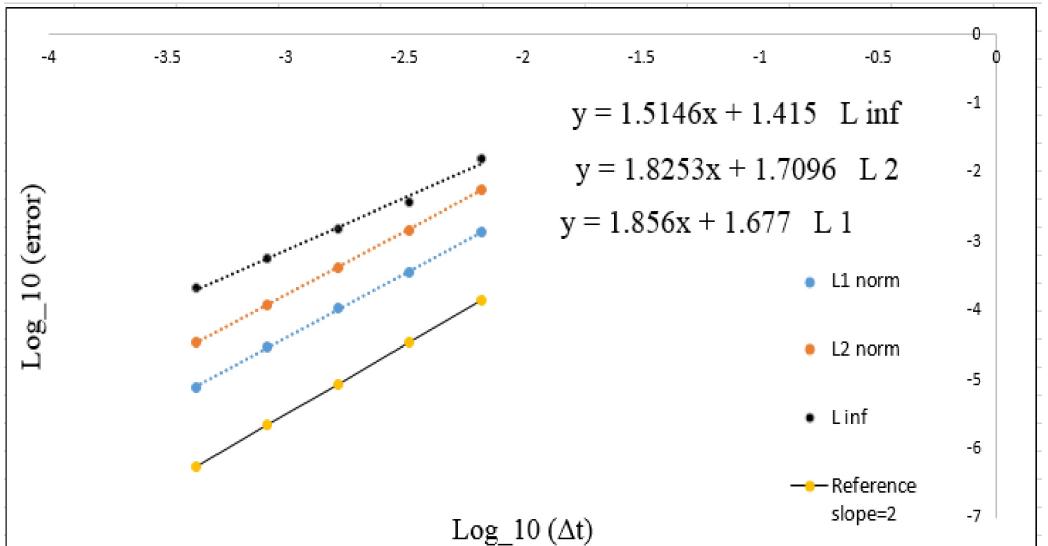
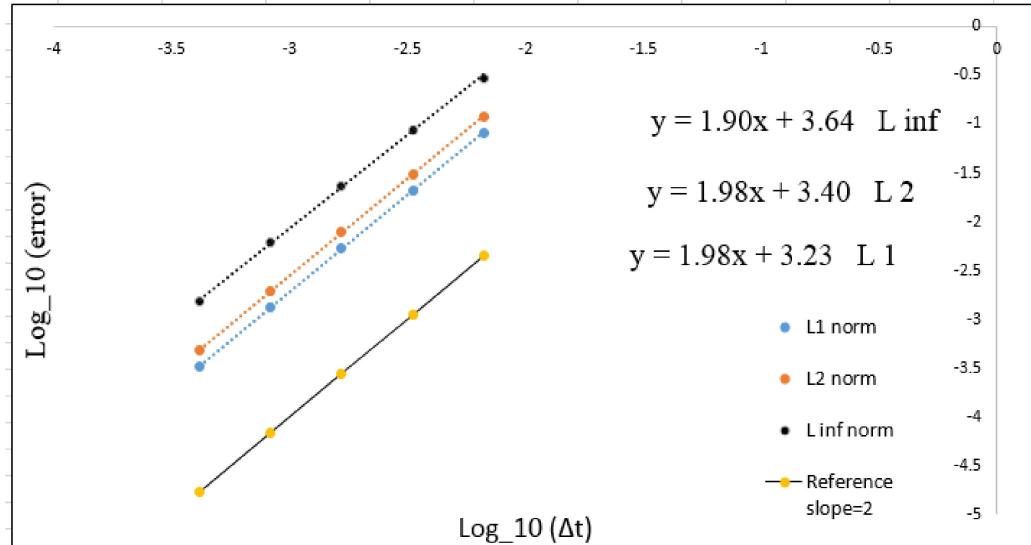
(b) Log-Log plot of Convergence rate for $\nabla \cdot \mathbf{u}^*$.

Figure 6.8: Plot of Convergence rates for Alg 2. Domain: $[-\frac{1}{2}, \frac{1}{2}]^2$, time = 1 and CFL = 0.5. In each plot, the data points corresponding to grid sizes of 15, 30, 60, 120, 240.



(a) Log-Log plot of Convergence rate for Pressure

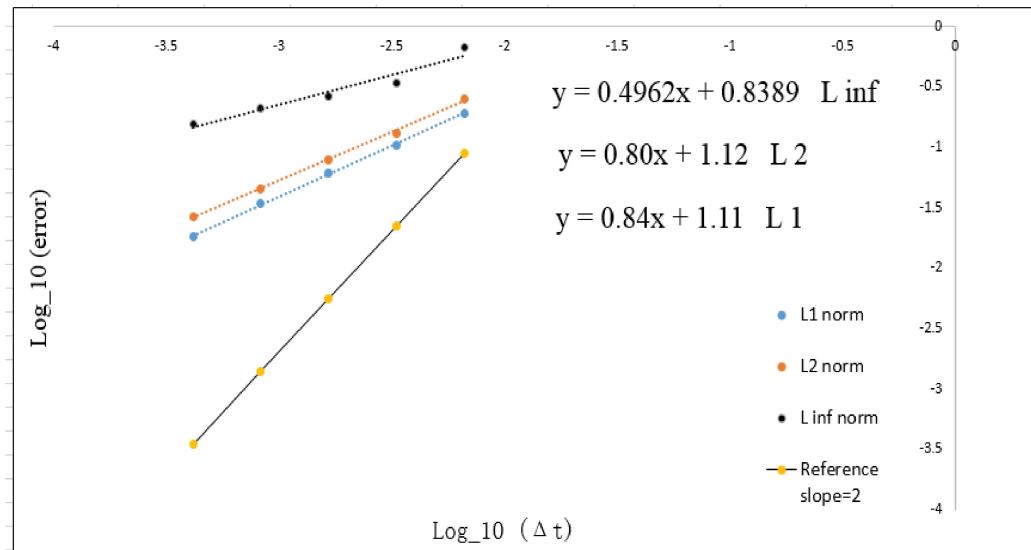
(b) Log-Log plot of Convergence rate for $\nabla \cdot \mathbf{u}^*$.

Figure 6.9: Plot of Convergence rates for Alg 3. Domain: $[-\frac{1}{2}, \frac{1}{2}]^2$, time = 1 and CFL = 0.5. In each plot, the data points corresponding to grid sizes of 15, 30, 60, 120, 240.

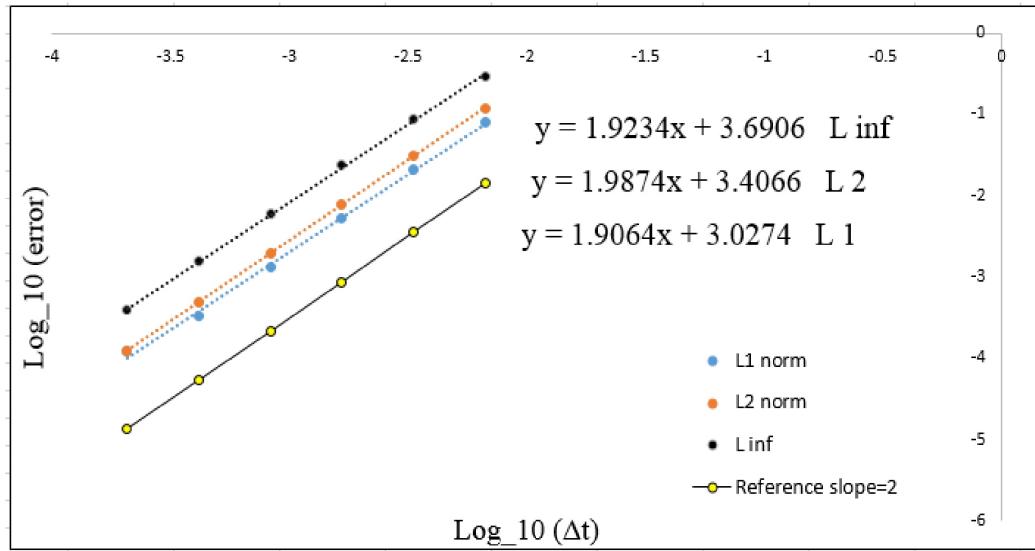
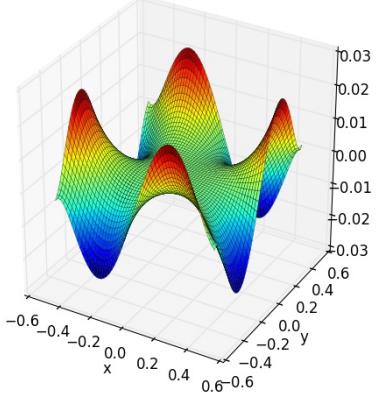
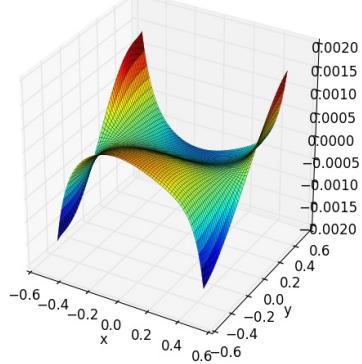


Figure 6.10: Long-Log plot of Pressure Convergence rates for Gauge method. Domain: $[-\frac{1}{2}, \frac{1}{2}]^2$, time = 1 and CFL = 0.5. The data points corresponding to grid sizes of 15, 30, 60, 120, 240, and 480.

Plot of Pressure Error field at Time = 1

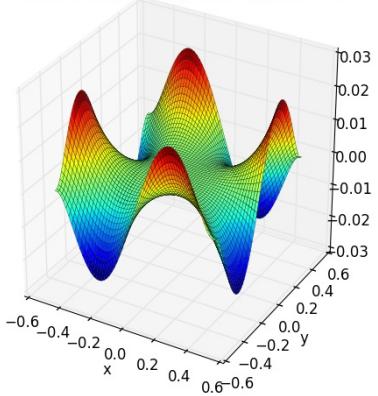


(a) Pressure error field for Alg 2 method

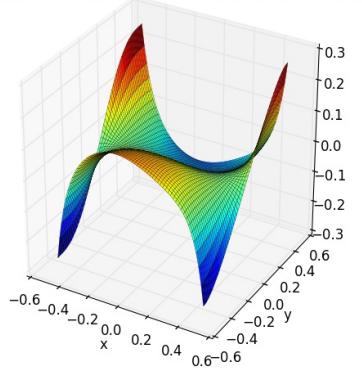
Plot of Divergence of intermediate velocity u^* at Time = 1

(b) Divergence of intermediate velocity field for Alg 2 method.

Plot of Pressure Error field at Time = 1

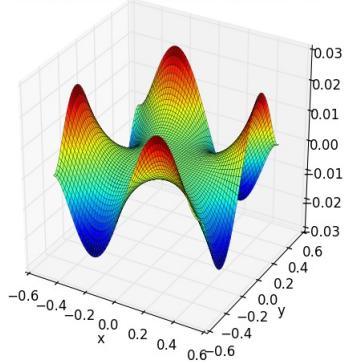


(c) Pressure error field for Alg 3 method

Plot of Divergence of intermediate velocity u^* at Time = 1

(d) Divergence of intermediate velocity field for Alg 3 method.

Plot of Pressure Error field at Time = 1



(e) Pressure error field for Gauge method.

Figure 6.11: Pressure error fields and divergence of intermediate velocity (if applicable) for Alg 2, Alg 3 and Gauge method at grid size of 120. Domain: $[-\frac{1}{2}, \frac{1}{2}]^2$, time = 1 and CFL = 0.5.

Necessity for accurate approximation to ϕ^{n+1} along the tangential boundary of the intermediate velocity field.

We investigate the importance of boundary condition choice for the intermediate velocity field and this might also help to understand why Alg 3 behaves better than Alg 2 in pressure convergence rates. We took two extrema here in choosing the tangential boundary condition for \mathbf{u}^* : one is use second order approximation to ϕ^{n+1} so we have $\tau \cdot \mathbf{u}^*|_{\partial\Omega} = \tau \cdot (\mathbf{u}^{n+1} + \nabla(2\phi^n - \phi^{n-1}))|_{\partial\Omega}$ (as of what we have done before for Alg 3); the other is to use no approximation to ϕ^{n+1} . We now rerun Alg 3 with the less accurate boundary where $\tau \cdot \mathbf{u}^* = \tau \cdot \mathbf{u}^{n+1}$ (this is the same as Alg 2). The convergence rates in pressure are again summarised in Figure below. It is observed that the rates are much reduced to about 0.7 order. The plot of pressure error field also shows four spikes at corners as in Alg 2. Hence this indicate the choice of tangential boundary condition to \mathbf{u}^* does indeed affect the performance of numerical boundary layer elimination. This finding indicates that for Alg 3 accurate approximation to ϕ^{n+1} must be implemented when computing the intermediate velocity field along the boundary points. This is consistent with the predication of our normal mode analysis. It is also consistent with the findings of others [6]. We infer that the same result might also work for Alg 2. However we did not have enough time to implement the different condition for Alg 2.

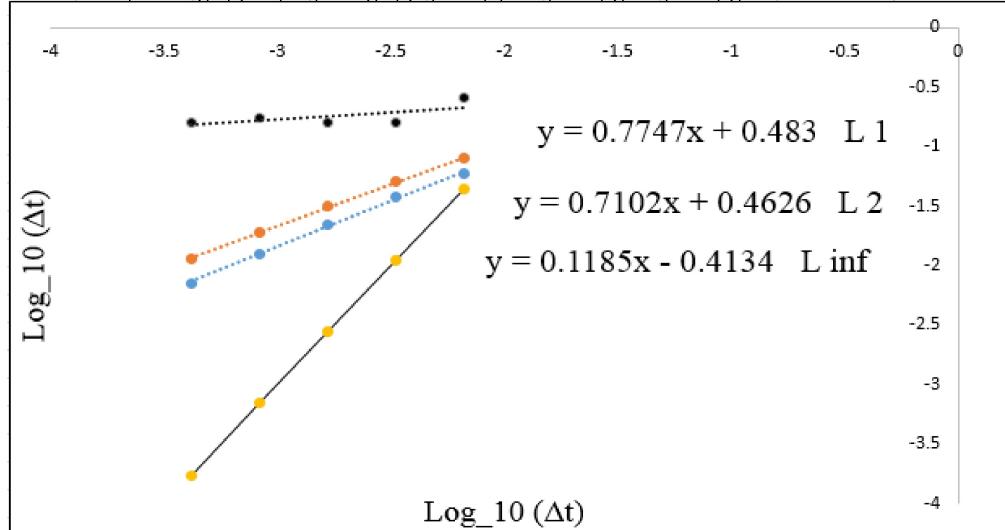
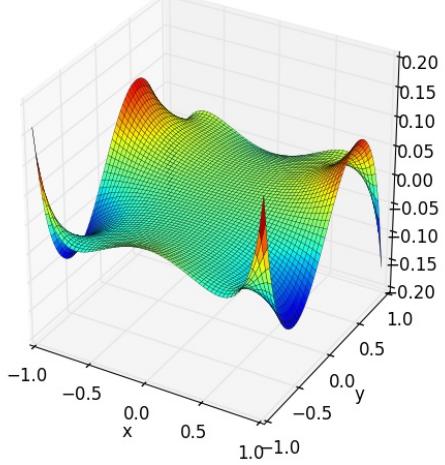
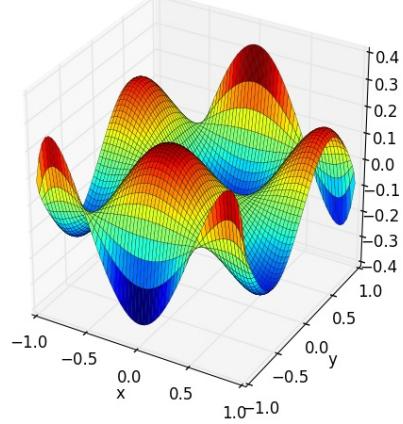


Figure 6.12: Log log plot of pressure error convergence rate for Alg 3 with less accurate boundary condition for $\nabla \cdot \mathbf{u}^*$.

Plot of Pressure Error field at Time = 1



(a) Pressure error field for Alg 3 with grid size 60 at time 1.

Plot of Divergence of intermediate velocity \mathbf{u}^* at Time = 1

(b) Plot of $\nabla \cdot \mathbf{u}^*$ for Alg 3 with grid size 60 at time 1.

Figure 6.13: Long-Log plot of Pressure Convergence rates and 3D surface plot of pressure error field for Alg 3 where the less accurate tangential boundary is used. Domain: $[-1, 1]^2$, time = 1 and CFL = 0.5. The data points corresponding to grid sizes of 15, 30, 60, 120, 240.

6.1.2 Unforced flow problem

To test the effect of smoothness of domain to the accuracy of projection methods. Another unforced flow problem is also considered. It is the well-known 2-D Taylor solution to the Navier Stokes equations. The analytic solutions are:

$$u = -\cos(x) \sin(y) e^{-2t} \quad (6.8a)$$

$$v = \sin(x) \cos(y) e^{-2t} \quad (6.8b)$$

$$p = -\frac{1}{4R}(\cos(2x) + \cos(2y))e^{-4t} \quad (6.8c)$$

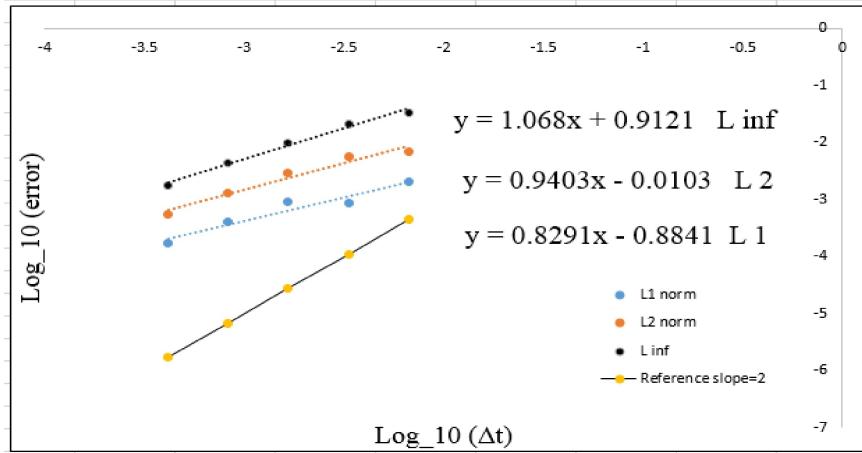
This test problem has non-zero boundary conditions for velocities and non-zero pressure gradients, unlike the forced flow example we have considered before where 2 of its boundaries are zero for velocities and pressure. Hence this should reveal more information on how the projection methods depends on the structure of domain and boundary conditions. For simplicity let's consider the domain: $[-\frac{\pi}{4}, \frac{\pi}{4}]^2$. Then the pressure value at these 4 corners ($x = \pm\frac{\pi}{4}$ and $y = \pm\frac{\pi}{4}$) would then be zero for all times which makes our pressure normalisation process easier.

In addition, this Taylor solutions are derived from the full Navier Stokes equations and hence it is also good to test our convective term solvers too.

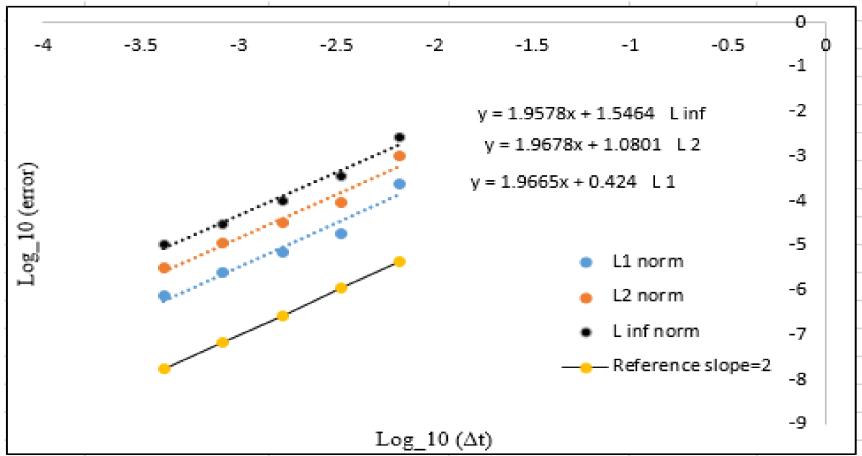
We have again run the Projection methods and Gauge for 5 grid sizes from 15×15 to 240×240 . The spatial stepping is halved each time. The solutions are calculated at time 1s and with Reynolds number equals to 1. A CFL number of 0.5 is used in all methods to obtain better error estimates.

The convergence rates in pressure are summarised in Figure. Interestingly all the methods show fully second order convergence except for Alg 2 where only first order convergence is observed. This time all the norms show degraded accuracy. Take a look at the Pressure error field we observe very large spikes located at the 4 corners of the domain. This indicates the presence of numerical boundary layers which are manifested better in the plot of Divergence of intermediate velocity field. Interestingly the magnitude of divergence of numerical boundary in intermediate velocity field is even larger for Alg 3 yet resulting in small Pressure error field. This result shows that the numerical boundary layers in ϕ and $\nabla \cdot \mathbf{u}^*$ cannot be fully filtered out in the projection methods unless accurate boundary condition for $\tau \cdot \mathbf{u}^*$ is implemented, same finding that supports the necessity for

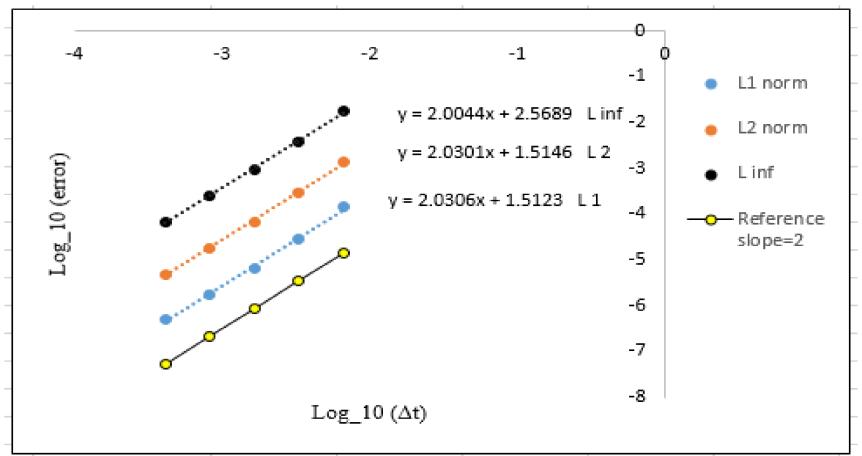
accurate tangential boundary condition for the intermediate velocity field. This is illustrated by the surface plots in Alg 2 where the thick boundary layer in $\nabla \cdot \mathbf{u}^*$ is eliminated in Pressure but the 4 spikes still present. This finding is also consistent with that obtained in David's paper [6].



(a) Log-Log plot of Convergence rate for Pressure Alg 2 method

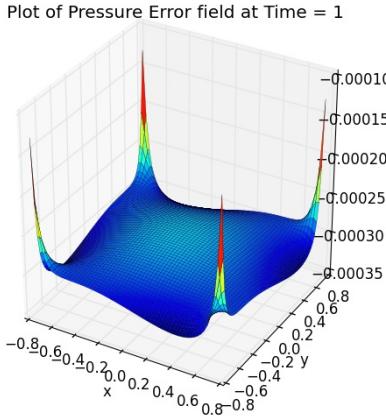


(b) Log-Log plot of Convergence rate for Pressure Alg 3.

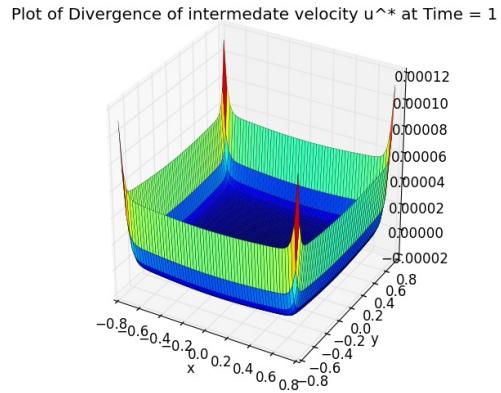


(c) Log-Log plot of Convergence rate for Pressure Gauge method

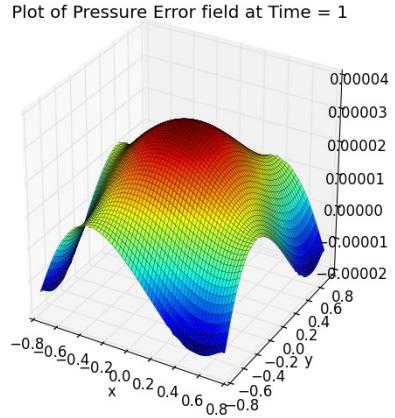
Figure 6.14: Plot of Convergence rates for the Unforced flow problem with Normalised Pressure approach used. Domain: $[-\frac{\pi}{4}, \frac{\pi}{4}]^2$, time = 1 and CFL = 0.5. In each plot, the data points corresponding to grid sizes of 15, 30, 60, 120, 240.



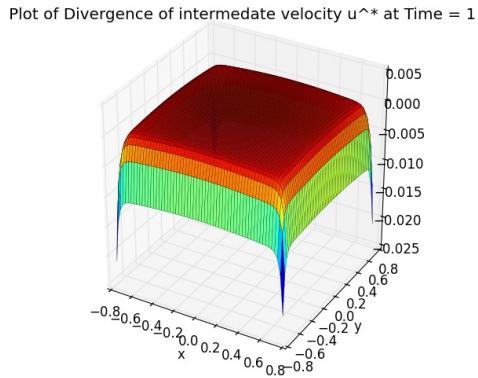
(a) Pressure error field for Alg 2 method



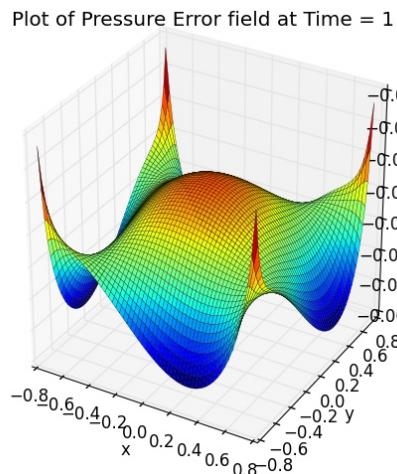
(b) Divergence of intermediate velocity field Alg 2



(c) Pressure error field for Alg 3 method



(d) Divergence of intermediate velocity field Alg 3



(e) Pressure error field Gauge method

Figure 6.15: Plot of Pressure error fields for the Unforced flow problem with Normalised Pressure approach used. Domain: $[-\frac{\pi}{4}, \frac{\pi}{4}]^2$, time = 1 and CFL = 0.5.

To further investigate the cause of degradation in accuracy in Alg 2, recall in normal mode analysis, we showed that the choice of pressure approximation $q = p^{n-1/2}$ results contradicting normal pressure gradient along the boundary especially if the normal analytic pressure gradient is not zero. This is the case in this unforced flow problem. For instance, the normal pressure gradient at west boundary ($x = \frac{\pi}{4}$ at any time is: $\frac{1}{2} \left(-\sin(\frac{\pi}{4}) \right)$) which is obviously not zero. Hence this means this choice of q does not work properly at the boundary.

It is supported by the plot of normal pressure gradient at West boundary. The plot of numerical solution is different to that of the analytical solution. This is amplified at the 4 corners of the domain where large spikes occur. We therefore infer that it is the non-smoothness caused by the inconsistent pressure approximation which degrades the global convergence of Alg 2.

We now propose a simplified modification to solve the problem by defining q to be $q = 2\phi^{n-1/2} - \phi^{n-3/2}$. This results in a second order accurate pressure approximation to $p^{n+1/2}$. This can be shown using a simple Taylor series argument. With this modified pressure approximation q , the numerical normal pressure gradient now approximates the analytic one better. It is converging to the analytic pressure gradient at higher than second order rate too. The non-smooth spikes are now eliminated which reduces the error and also lifts up the global pressure error convergence to 2.5 order.

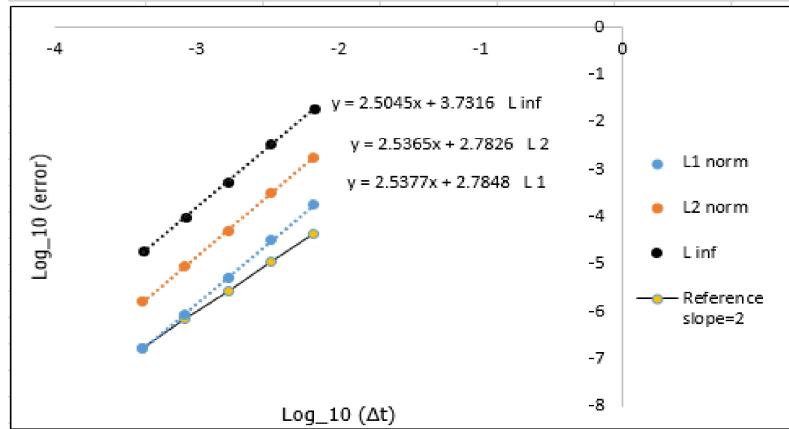
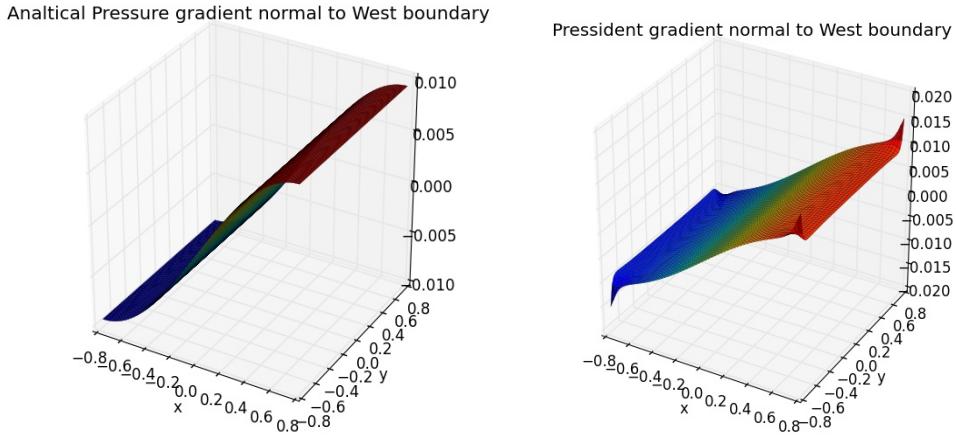


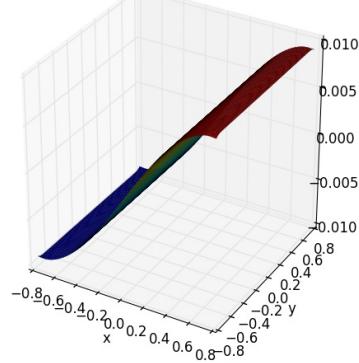
Figure 6.16: Log - log plot of the pressure error convergence rate with the modified $q = 2\phi^{n-1/2} - \phi^{n-3/2}$ used



(a) Plot of analytic pressure gradient normal to west boundary

(b) Plot of numerical pressure gradient normal to west boundary with $q = p^{n-1/2}$ is used

Numerical Pressident gradient normal to West boundary



(c) Plot of numerical pressure gradient normal to west boundary with the modified $q = 2\phi^{n-1/2} - \phi^{n-3/2}$ used

Figure 6.17: Plot of normal pressure gradients and convergence rates for Alg 2 with modified $q = 2\phi^{n-1/2} - \phi^{n-3/2}$ used. Domain: $[-\frac{\pi}{4}, \frac{\pi}{4}]^2$, time = 1 and CFL = 0.5.

6.2 Driven cavity flow

In this section, we consider a very interesting flow problem: the Lid-Driven cavity. Driven cavity flow has been well studied in the literature. It is a simple example to show the effect of Reynolds number and turbulent and unsteady flows. We mainly illustrate the qualitative findings here through numerical simulations. There are however benchmark results for 3D and 2D Lid-driven cavity problems for the purpose of comparing accuracy of numerical solvers.

(**citation, Goda, Shen...**)

Although the problem set up could vary, however the essential idea is the same. We have a fluid initially at rest except at one boundary where either horizontal or vertical component has a constant non-zero velocity (i.e. 1). The flow in the interior is then driven by the non-zero velocity at the boundary. This is like opening the lid along the non-zero boundary and sliding it at a constant velocity. The flow pattern depend on the value of Reynolds numbers and with low to moderate Reynolds number (often below 10000) the flow can reach to steady state whereas for higher Reynolds numbers, turbulence will occur in which no steady state solutions can be obtained. In fact, turbulence only occur in 3-dimensional type of flows and hence our 2D numerical solver would not show the pattern completely. Nevertheless, as our result presents, as Reynolds number increases, the flow does become unsteady which is aligned with the turbulence behaviour in 3D simulations.

6.2.1 Problem set up

We consider a fluid confined in the square domain: $[0, 1]^2$. The fluid velocities are initially zero except for v at the East boundary ($x = 1$) which it is equal to 1 (i.e. $v(x = 1, y, 0) = 1$). The problem set up is illustrated in the diagram below. The full Navier Stokes equations of incompressible flow is used. We consider the flow at different Reynolds numbers ($R = 1000, 10000$). The Gauge method was used to compute the solutions with grid size 100×100 and $CFL = 0.5$.

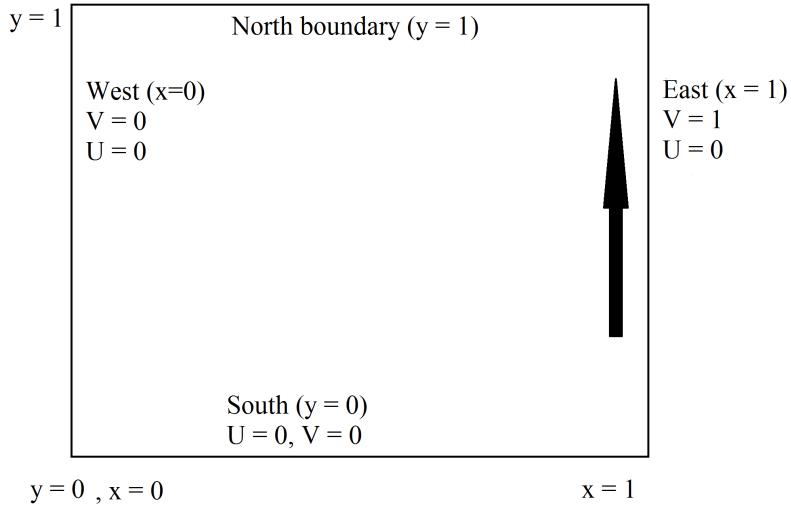
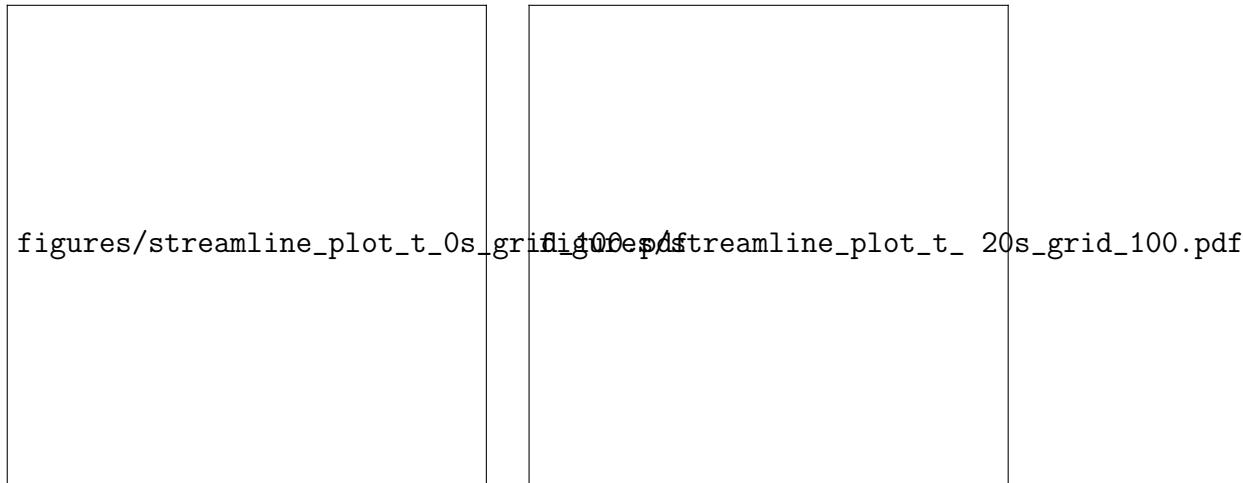


Figure 6.18: Problem setup of the Lid-driven cavity. The flow is driven by the constant vertical velocity along the East boundary ($v = 1$ at $x = 1$)

First we consider moderate Reynolds number flows of with value of 1000, the fluid converges to the steady state after about a time of 20s. 3 main eddies are formed. The main one starts from the corner near the East boundary since the initial flow starts from here. It then slides to the middle over time forming a constant main vortex. Mean while two other smaller eddies are then formed one after another. The streamline plots are presented below illustrate the flow patterns.

6.2.2 Results

At higher Reynolds number the flow become unsteady (or “turbulent”) as more smaller unsteady eddies are formed. This is clearly shown by the streamline plot at Reynold = 10000. Hence our result is in line with the literature too.



(a) Streamline plot of velocity fields at time 0s (b) Streamline plot of velocity fields at time 20s

Figure 6.19: Streamline plot of velocity fields for driven cavity flow. $v(x = 1, y, t) = 1$ with grid size 100 and CFL = 0.5 used.

Reynolds number = 10000

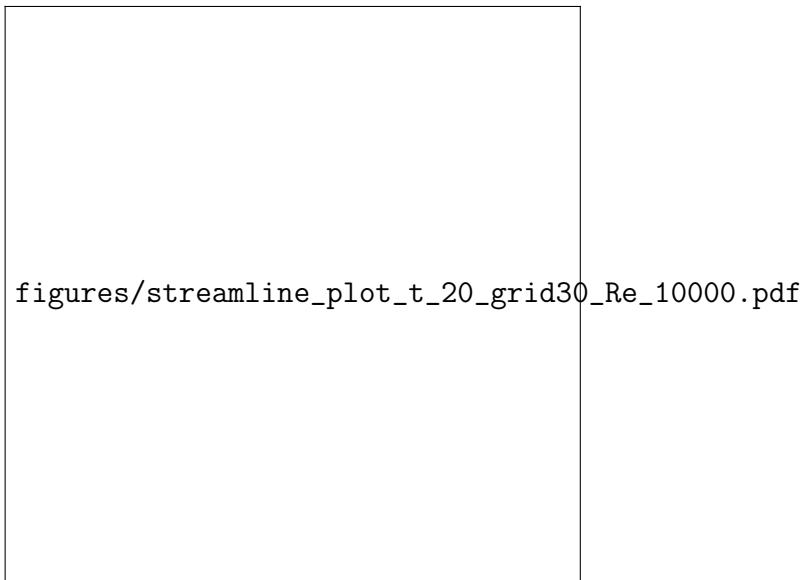


Figure 6.20: Streamline plot of velocity fields at time 20s with Reynolds number equal to 10000

Plots of velocity and pressure solutions are also shown below:

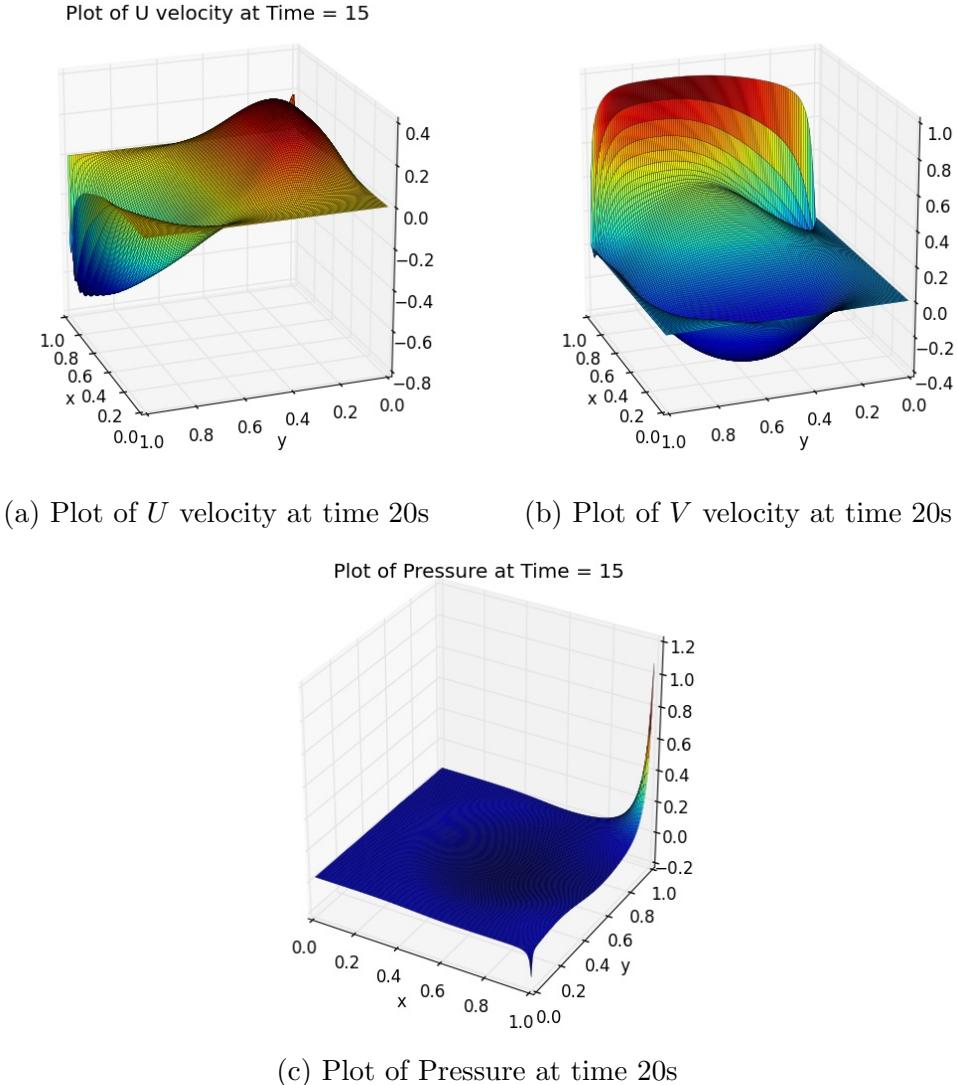


Figure 6.21: Plot of velocity and pressure solutions for driven cavity problem with Reynolds number equal to 1000. The East boundary for V velocity is kept to be 1 over time. Grid size of 100 and CFL = 0.5 was used.

Although our result shows the unsteady pattern of flow as Reynolds number increases which is in line with the literature in general **Citation**, however to examine the turbulent behaviour we should implement 3D numerical solvers. This could be a further investigation in the future. Once again this section is only aimed to be a qualitative illustration of the classic Lid-driven cavity problem.

Chapter 7

Conclusions

In this thesis, we have examined robust Projection method to the numerical solutions of Incompressible Navier Stokes equations. We have talked about the origin of the method which is based on the idea of Helmholtz Hodge decomposition theorem. We have described the Chorin's original method and the second order extensions as well as the Gauge method. We have analysed these methods' accuracy through normal mode analysis. In a periodic channel domain,(following Brown's analysis) our normal mode analysis is predicting second order accuracy for all projection methods. This is also consistent with our numerical results. However further analysis could be done in a more general domain because normal mode analysis is restricted on the domain we work with and the error convergence might behave differently too [24].

In summary:

We observe that the Gauge method shows fully 2nd order convergence in Pressure in all examples and domains considered whereas Projection methods show a degraded accuracy in some examples (see Unforced flow example in Results chapter). We observe that the Projection methods depends strongly on the smooth of domain (especially for Alg 2 with a lagged pressure approximation used). The exact cause of this problem still remains open ([15]). This finding is consistent with Shen et.al where they have bounded the L_2 norm of Pressure error to be only 1.5 order accurate in general domains (e.g. the square domain with Dirichlet boundary conditions we have considered in the Forced flow example). For details of the proof see [15, 24]. In the case of periodic channel, all projection methods show 2nd order accuracy in pressure. This is consistent with the predication of error estimates by our normal mode analysis done in previous

chapter where a periodic channel domain was considered.

Necessity for accurate boundary condition of the tangential component of the intermediate velocity field (\mathbf{u}^*). We have observed that second order approximation ϕ^{n+1} is needed when computing $\tau \cdot \mathbf{u}^*$ in the projection step. This is consistent with our normal mode analysis predictions and findings of other researchers too [6].

Modification to pressure approximation in Alg 2 We have observed that the inconsistent normal pressure gradient in Alg 2 is caused by an inappropriate choice of pressure approximation q used. This introduces non-smooth along the boundary of pressure gradients. This is more evident when the test problems have non-zero pressure gradients. However with a modified $q = 2\phi^{n-1/2} - \phi^{n-3/2}$, we restore fully second order convergence in pressure. The exact reason of this improvement however still subject to more careful considerations.

Further research could be looking at flow in general domain and complex geometry, Interface problems. Finite element could also to be used too. We are also interested in higher order schemes e.g. 4th 6th order schemes using compact finite difference.

Bibliography

- [1] A. S. Almgren, J. B. Bell, and W. Y. Crutchfield. Approximate projection methods: Part i. inviscid analysis. *SIAM Journal on Scientific Computing*, 22(4):1139–1159, 2000.
- [2] A. S. Almgren, J. B. Bell, and W. G. Szymczak. A numerical method for the incompressible navier-stokes equations based on an approximate projection. *SIAM Journal on Scientific Computing*, 17(2):358–369, 1996.
- [3] G. B. Arfken and H. J. Weber. *Mathematical Methods For Physicists International Student Edition*. Academic press, 2005.
- [4] S. Armfield and R. Street. Fractional step methods for the navier-stokes equations on non-staggered grids. *ANZIAM Journal*, 42:C134–C156, 2000.
- [5] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 85(2):257–283, 1989.
- [6] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible navier–stokes equations. *Journal of Computational Physics*, 168(2):464–499, 2001.
- [7] A. J. Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.
- [8] A. J. Chorin. On the convergence of discrete approximations to the navier–stokes equations. *Mathematics of Computation*, 23(106):341–353, 1969.
- [9] A. J. Chorin and J. E. Marsden. *A mathematical introduction to fluid mechanics*, volume 3. Springer, 1990.
- [10] R. Courant and D. Hilbert. *Methods of mathematical physics*, volume 1. CUP Archive, 1966.

- [11] O. Darrigol. Between hydrodynamics and elasticity theory: the first five births of the navier-stokes equation. *Archive for history of exact sciences*, 56(2):95–150, 2002.
- [12] H. Fujita and T. Kato. On the navier-stokes initial value problem. i. *Archive for rational mechanics and analysis*, 16(4):269–315, 1964.
- [13] K. Goda. A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows. *Journal of Computational Physics*, 30(1):76–95, 1979.
- [14] J. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Computer methods in applied mechanics and engineering*, 195(44):6011–6045, 2006.
- [15] J. Guermond and J. Shen. On the error estimates for the rotational pressure-correction projection methods. *Mathematics of Computation*, 73(248):1719–1737, 2004.
- [16] F. H. Harlow, J. E. Welch, et al. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids*, 8(12):2182, 1965.
- [17] L. H. Howell and J. B. Bell. An adaptive mesh projection method for viscous incompressible flow. *SIAM Journal on Scientific Computing*, 18(4):996–1013, 1997.
- [18] J. Kim and P. Moin. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics*, 59(2):308–323, 1985.
- [19] M. F. Lal, M. F. Lai, M. Fruchtmann, et al. A projection method for reacting flow in the zero mach number limit. 1993.
- [20] J.-G. Liu. Projection method ii: Godunov-ryabenki analysis. *SIAM journal on numerical analysis*, 33(4):1597–1621, 1996.
- [21] F. Maria Denaro. On the application of the helmholtz–hodge decomposition in projection methods for incompressible flows with general boundary conditions. *International Journal for Numerical Methods in Fluids*, 43(1):43–69, 2003.

- [22] M. L. Minion. A projection method for locally refined grids. *Journal of Computational Physics*, 127(1):158–178, 1996.
- [23] J. B. Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108(1):51–58, 1993.
- [24] J.-H. Pyo and J. Shen. Normal mode analysis of second-order projection methods for incompressible flows. *DISCRETE AND CONTINUOUS DYNAMICAL SYSTEMS SERIES B*, 5(3):817, 2005.
- [25] R. Rannacher. *On Chorin’s projection method for the incompressible Navier-Stokes equations*. Springer, 1992.
- [26] J. Shen. On error estimates of projection methods for navier-stokes equations: first-order schemes. *SIAM Journal on Numerical Analysis*, 29(1):57–77, 1992.
- [27] J. Shen. On error estimates of the projection methods for the navier-stokes equations: second-order schemes. *Mathematics of Computation of the American Mathematical Society*, 65(215):1039–1065, 1996.
- [28] J. C. Strikwerda and Y. S. Lee. The accuracy of the fractional step method. *SIAM Journal on numerical analysis*, 37(1):37–47, 1999.
- [29] R. Temam. Sur l’approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (ii). *Archive for Rational Mechanics and Analysis*, 33(5):377–385, 1969.
- [30] R. Temam. Remark on the pressure boundary condition for the projection method. *Theoretical and Computational Fluid Dynamics*, 3(3):181–184, 1991.
- [31] R. Temam. *Navier-Stokes equations and nonlinear functional analysis*, volume 66. Siam, 1995.
- [32] J. Van Kan. A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing*, 7(3):870–891, 1986.
- [33] E. Weinan, J.-G. Liu, et al. Gauge method for viscous incompressible flows. *Communications in Mathematical Sciences*, 1(2):317–332, 2003.

- [34] J.-F. Wong and M.-H. Chan. A consistent splitting scheme for unsteady incompressible viscous flows i. dirichlet boundary condition and applications. *International journal for numerical methods in fluids*, 51(4):385–424, 2006.