

# NUMERICAL SOLUTIONS OF THE 2-D NAVIER-STOKES EQUATIONS

by

Marvin Washington

A Thesis

Submitted to the Division of Graduate Studies

Jackson State University

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

May 2011

Major Subject: Mathematics

# NUMERICAL SOLUTIONS OF THE 2-D NAVIER-STOKES EQUATIONS

A Thesis

by

Marvin Washington

Approved:

---

Committee Chairperson

Shaungzhang Tu, PhD

---

Committee Member, Advisor

Célestin Wafo Soh, PhD

---

Committee Member

Marvin D. Watts, PhD

---

Dean, Division of Graduate Studies

Dorris R. Robinson-Gardner, PhD

May 2011

# ABSTRACT

Navier-Stokes Equations are nonlinear partial differential equations governing the dynamics of fluids. They arise in several areas of practical importance such as meteorology, aerodynamics, geophysics, and animation in movies or video games. They comprise two set of equations encoding conservation of linear momentum and mass. These equations must be supplemented with appropriate constitutive relations and boundary conditions. The former provides a relation between stress and rate of shear (strain). They allow the classification of fluids into Newtonian and non-Newtonian fluids. Newtonian fluids are characterized by a linear relation between stress and strain whereas non-Newtonian fluids violate such a relation. This thesis, will deal primarily with Newtonian fluids which are incompressible i.e. their density (mass per unit volume) is constant. Boundary conditions such as the no-slip boundary conditions (Dirichlet boundary conditions), Neumann boundary conditions, and periodic boundary conditions will be employed. The existence, uniqueness, and regularity of solutions of the Navier-Stokes equations under appropriate boundary conditions and data are well-known in 2-D.

In 3-D there are still unsolved problems related to existence, uniqueness, and regularity of solutions of Navier-Stokes Equations. The main concern in this thesis will be the numerical solution of the 2-D Navier-Stokes Equations under various assumptions on the boundary conditions and the geometry. The finite difference method will be used in all the numerical schemes and the validity of the numerical techniques used will be checked by simulating some classical flows. The work will be organized as follows. There will be six chapters. In the first chapter, a rigorous derivation of the Navier-Stokes equations is provided. Also, boundary conditions and the equation satisfied by pressure are discussed. In chapter two, the discretization of the 2-D Navier-Stokes Equations by finite-differencing is dealt with. In particular, methods for enforcing boundary conditions and the incompressibility conditions are elaborated on. Concerning the later, Chorin's Projection Method is described in detail. In chapter three, the implementation of the numerical scheme described in the previous chapter is supplied. In chapter four, flow visualization techniques are discussed and implemented. In chapter five, some classical flows are examined. In the final chapter, the work is summarized.

# DEDICATION

*To my Family and my Friends*

# Contents

<b>Marvin</b>	<b>1</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Derivation of Navier-Stokes Equations</b>	<b>3</b>
2.1 Reynold's transport theorem . . . . .	4
2.2 Conservation of Mass . . . . .	11
2.3 Conservation of Momentum . . . . .	13
2.4 Dimensional Analysis . . . . .	15

2.5	Geometries and Boundary Conditions . . . . .	17
2.5.1	Dirichlet Boundary Condition . . . . .	17
2.5.2	Neumann Boundary Condition . . . . .	18
2.5.3	Mixed Boundary Condition . . . . .	19
2.5.4	Periodic Boundary Condition . . . . .	19
2.6	Conclusion . . . . .	20
<b>3</b>	<b>Discretization of the Navier-Stokes Equations</b>	<b>21</b>
3.1	Finite Difference Approximations . . . . .	22
3.1.1	First Order Finite Difference Approximations . . . . .	22
3.1.2	Higher Order Finite Difference Approximations . . . . .	25
3.2	The Donor-Cell Discretization Scheme . . . . .	27
3.3	Discretization of the Heat Equation . . . . .	28
3.4	Discretization of the Spatial Derivative . . . . .	29

3.5	Discretization of the Time Derivative . . . . .	33
3.5.1	The Time-Stepping Algorithm . . . . .	34
3.6	Conclusion . . . . .	38
<b>4</b>	<b>Implementation of the Navier-Stokes Equations Solver</b>	<b>39</b>
4.1	The Program . . . . .	40
4.2	Conclusion . . . . .	44
<b>5</b>	<b>Visualization Techniques</b>	<b>45</b>
5.1	Vector Field Representation . . . . .	46
5.2	Streamlines . . . . .	46
5.3	Conclusion . . . . .	46
<b>6</b>	<b>Examination of Classical Flows</b>	<b>48</b>
6.1	Flow Over a Flat Plate . . . . .	48



6.2	Flow Between Two Parallel Plates . . . . .	50
6.3	Lid-Driven Cavity Flow . . . . .	53
6.4	Flow Over a Backward Facing Step . . . . .	53
6.5	Conclusion . . . . .	55
<b>7</b>	<b>Summary</b>	<b>58</b>
	<b>References</b>	<b>60</b>

## ACKNOWLEDGEMENTS

I am particularly grateful to my supervisor Dr. Celestin Wafo Soh for his unfailing support throughout the writing of this thesis. His multiple advices and suggestions have considerably enhanced the readability of earlier drafts. Foremost, his enthusiasm and dedication to research have been a great source of inspiration.

I have also benefited from the most needed moral support of my family and friends. Their phone calls and letters have kept me from stress and its consequences.

I would like to express my sincere appreciation to Dr. Shaungzhang Tu and Dr Marvin Watts for agreeing to be apart of my committee an also for their guidance in the preparation of this thesis.

Finally, I am very much indebted to Jackson State University for its financial assistance through the tuition assistance recieved from the Department of Mathematics and Dr. Hyun J. Cho for hiring me as a research assistant for an Enviornmental Science Project.

# Chapter 1

## Introduction

Navier-Stokes Equations, named after Claude-Louis Navier and George Gabriel Stokes, are nonlinear partial differential equations governing the dynamics of fluids. They arise in several areas of practical importance such as meteorology, aerodynamics, geophysics, and animation in movies or video games. They comprise a set of equations encoding conservation of mass and linear momentum. These equations must be supplemented with appropriate constitutive relations and boundary conditions. The former provides a relation between stress and rate of shear (strain). They allow the classification of fluids into Newtonian and non-Newtonian fluids. Newtonian fluids are characterized by a linear relation between

stress and strain whereas non-Newtonian fluids violate such a relation. The existence, uniqueness, and regularity of solutions of the Navier-Stokes equations under appropriate boundary conditions and data are well-known in 2-D. In 3-D there are still unsolved problems related to existence, uniqueness, and regularity of solutions of Navier-Stokes Equations. Arriving at the proper set of equations to model the behavior of different fluids required get diligence and hard work from individuals such as Leonhard Euler, Augustin-Louis Cauchy, and Simon Denis Poisson to name a few.

## Chapter 2

# Derivation of Navier-Stokes Equations

In this chapter, we obtain Navier-Stokes Equations from the conservation of mass and linear momentum. The main ingredient in our derivation is Reynold's transport theorem. The latter is a generalization of the classical Leibniz formula. It permits the computation of the time rate of change of an integral on a time-dependent domain. The analysis preceding the derivation of the Navier-Stokes equations relies on the lecture notes by Mason [5, 6].

## 2.1 Reynold's transport theorem

Consider a fluid body which is composed of a set of particles. At each instant  $t$ , the particles are assigned a unique point in a closed and bounded region  $\mathcal{B}_t$  of the three-dimensional Euclidean space. Each point of  $\mathcal{B}_t$  is occupied by only one particle.

**Definition 2.1**  $\mathcal{B}_t$  is called the **configuration** of the body at time  $t$ .

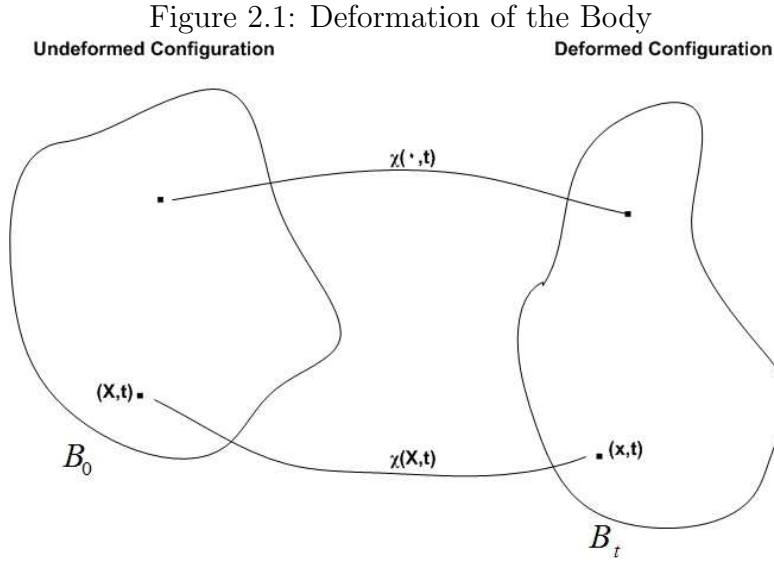
The configuration at time  $t = 0$  is usually chosen as the reference configuration and denoted by  $\mathcal{B}_0$ . Let  $\mathbf{X} = (X_1, X_2, X_3)$  be the Cartesian coordinates of a particle in  $\mathcal{B}_0$ . The motion of the body is described by giving the position  $\mathbf{x}$  of the particle  $\mathbf{X}$  at time  $t$ .

Let  $\chi : \Omega_1 \times (0, T) \longrightarrow \Omega_2$ ; where  $(\mathbf{X}, t) \longmapsto \mathbf{x} = \chi(\mathbf{X}, t) = (\chi_1(X, t), \chi_2(X, t), \chi_3(X, t))$

$\chi_i, (i = 1, 2, 3)$  are differentiable with continuous derivatives.

$\forall t \in (0, T), \chi(\cdot, t) : \Omega_1 \longrightarrow \Omega_2$  is a diffeomorphism.

**Definition 2.2** The mapping  $\chi$  from the initial configuration to the final configuration is called a **deformation** of the body.



The coordinates of the point  $\mathbf{X}$ , denoted by  $(X_1, X_2, X_3)$  are called material(Lagrangian) coordinates.

The coordinates of the point  $\mathbf{x}$ , denoted by  $(x_1, x_2, x_3)$  are called spatial(Eulerian) coordinates.

Figure (2.1) illustrates the deformation of a body from its undeformed configuration to its deformed configuration.

We shall adopt the following notations:

Capital letters A,B,C... as indices refer to material coordinates:  $X_A$ ,  $A \in \{1, 2, 3\}$

Lowercase letters i,j,k... as indices refer to spatial coordinates:  $x_i$ ,  $i \in \{1, 2, 3\}$

Einstein's summation convention:

$$x_i x_i = x_1^2 + x_2^2 + x_3^2$$

$$X_A X_A = X_1^2 + X_2^2 + X_3^2$$

**Definition 2.3** The ***velocity*** of the particle  $X$  denoted  $\dot{x}$  is defined by

$$\boldsymbol{v} = \dot{\boldsymbol{x}} = \frac{\partial \chi(\boldsymbol{X}, t)}{\partial t} \quad (2.1)$$

**Definition 2.4** The ***acceleration*** of the particle  $X$  denoted  $\ddot{x}$  is defined by

$$\boldsymbol{v} = \ddot{\boldsymbol{x}} = \frac{\partial^2 \chi(\boldsymbol{X}, t)}{\partial t^2} \quad (2.2)$$

A quantity can be expressed as a function of either the Lagrangian description  $(\boldsymbol{X}, t)$  or the Eulerian description  $(\boldsymbol{x}, t)$ . Changing from one description to the other can be done by using

$$\Psi(\chi^{-1}(\boldsymbol{x}, t), t) = \psi(\boldsymbol{x}, t)$$

or

$$\psi(\chi(\boldsymbol{X}, t), t) = \Psi(\boldsymbol{X}, t)$$



**Definition 2.5** The **alternating tensor**, denoted as  $\epsilon_{ABC}$ , is defined by

$$\epsilon_{ABC} = \begin{cases} 1 & \text{if } (ABC) \text{ is an even permutation of } \{1, 2, 3\} \\ -1 & \text{if } (ABC) \text{ is an odd permutation of } \{1, 2, 3\} \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

**Definition 2.6** The **deformation gradient** denoted  $F_\chi$  is defined by

$$F_\chi = x_{i,A} = \left[ \frac{\partial \chi_i}{\partial X_A} \right] i, A \in \{1, 2, 3\} \quad (2.4)$$

The deformation gradient of the map  $\chi$  is non-singular, since  $\chi$  is invertible, and its Jacobian is defined as  $J_\chi = \det F_\chi$ .

The meaning of  $J_\chi$  as the change in volume due to the change of coordinates  $\chi$  can be easily understood from the following theorem.

**Theorem 2.1** Let  $dV$  be an infinitesimal domain, given by the triple product  $dV = dx^{(1)} \cdot (dx^{(2)} \times dx^{(3)})$ . Also let  $dV_0 = dX^{(1)} \cdot (dX^{(2)} \times dX^{(3)})$  be the push forward of  $dV$ , that is  $dx^{(i)} = F_\chi dX^{(i)}$ , for  $i \in \{1, 2, 3\}$ . Then,

$$dV = J_\chi dV_0.$$

**Proof.**

$$dx_i^{(\alpha)} = x_{i,A} dX_A^{(\alpha)}$$

Thus

$$\begin{aligned}
 \det[dx_i^{(\alpha)}] &= \det[x_{i,A} dX_A^{(\alpha)}] \\
 &= \det[x_{i,A}] \det[dX_A^{(\alpha)}] \\
 &= J_\chi \det[dX_A^{(\alpha)}]
 \end{aligned}$$

But,

$$\begin{aligned}
 \det[dx_i^{(\alpha)}] &= \epsilon_{ijk} dx_i^{(1)} dx_j^{(2)} dx_k^{(3)} \\
 &= dx_i^{(1)} \epsilon_{ijk} dx_j^{(2)} dx_k^{(3)} \\
 &= dX^{(1)} \cdot (dX^{(2)} \times dX^{(3)})
 \end{aligned}$$

Therefore,

$$d\vec{x} = J_\chi d\vec{X} .$$

which concludes the proof.

**Theorem 2.2**

$$\frac{DJ_\chi}{Dt} = J_\chi \operatorname{div}(v)$$

**Proof.**

$$J_\chi = \det[x_{i,A}] = \epsilon_{ABC} x_{1,A} x_{2,B} x_{3,C}$$

Thus

$$\frac{DJ_\chi}{Dt} = \epsilon_{ABC} \left[ \left( \frac{D}{Dt} x_{1,A} \right) x_{2,B} x_{3,C} + x_{1,A} \left( \frac{D}{Dt} x_{2,B} \right) x_{3,C} + x_{1,A} x_{2,B} \left( \frac{D}{Dt} x_{3,C} \right) \right]$$

But,

$$\begin{aligned} \frac{D}{Dt} x_{1,A} &= \frac{\partial^2}{\partial t \partial X_A} x_1(X, t) \\ &= \frac{\partial^2}{\partial X_A \partial t} x_1(X, t) \\ &= \frac{\partial v_1}{\partial X_A} \\ &= \frac{\partial v_1}{\partial x_k} \frac{\partial x_k}{\partial X_A} \\ &= v_{1,k} \frac{\partial x_k}{\partial X_A} \end{aligned}$$

Thus

$$\begin{aligned} \epsilon_{ABC} \left( \frac{D}{Dt} x_{1,A} \right) x_{2,B} x_{3,C} &= \epsilon_{ABC} v_{1,k} x_{k,A} x_{2,B} x_{3,C} \\ &= v_{1,1} \epsilon_{ABC} x_{1,A} x_{2,B} x_{3,C} \\ &= v_{1,1} J_\chi \end{aligned}$$

Similarly,

$$\epsilon_{ABC}x_{1,A} \left(\frac{D}{Dt}\right)x_{2,B}x_{3,C} = v_{2,2}J_\chi$$

and

$$\epsilon_{ABC}x_{1,A}x_{2,B}\left(\frac{D}{Dt}\right)x_{3,C} = v_{3,3}J_\chi$$

Thus

$$\begin{aligned} \frac{DJ_\chi}{Dt} &= v_{k,k}J_\chi \\ &= J_\chi \operatorname{div}(v) \end{aligned}$$

**Theorem 2.3** *Let  $V$  be a subset of the deformed configuration, and let  $V_0$  be a subset of the undeformed configuration. Then for a differentiable function  $f$  with continuous partial derivatives, we have*

$$\frac{D}{Dt} \int_V f(\vec{x}, t) dV = \int_V \left\{ \frac{\partial f}{\partial t} + \nabla \cdot (f \vec{u}) \right\} dV \quad (2.5)$$

**Proof.**

$$\begin{aligned} \frac{D}{Dt} \int_V f(\vec{x}, t) dV &= \frac{D}{Dt} \int_{V_0} f(\chi(\mathbf{X}, t), t) J_\chi dV_0 \\ &= \int_{V_0} \left\{ \frac{Df(\chi(\mathbf{X}, t), t)}{Dt} J_\chi + f(\chi(\mathbf{X}, t), t) \frac{DJ_\chi}{Dt} \right\} dV_0 \end{aligned}$$

$$\begin{aligned}
&= \int_{V_0} \left\{ \frac{Df(\chi(\mathbf{X}, t), t)}{Dt} J_\chi + f(\chi(\mathbf{X}, t), t) J_\chi \nabla v \right\} dV_0 \\
&= \int_{V_0} \left\{ \frac{Df(\chi(\mathbf{X}, t), t)}{Dt} + f(\chi(\mathbf{X}, t), t) \nabla v \right\} J_\chi dV_0 \\
&= \int_{V_0} \left\{ \frac{\partial f(\chi(\mathbf{X}, t), t)}{\partial t} + v \cdot \nabla f(\chi(\mathbf{X}, t), t) + f(\chi(\mathbf{X}, t), t) \cdot \nabla v \right\} J_\chi dV_0 \\
&= \int_V \left\{ \frac{\partial f(\vec{x}, t)}{\partial t} + \operatorname{div}(f(\vec{x}, t)v) \right\} dV
\end{aligned}$$

The Dominated Convergence Theorem is what allows for the passing of the material derivative inside of the integral operator.

## 2.2 Conservation of Mass

The mass of a fluid ( $M_{fluid}$ ) is given by the integral of the fluid density over the fluid domain i.e.

$$M_{fluid} = \int_V \rho(\vec{x}, t) dV \quad (2.6)$$

The Conservation of Mass principle states that the derivative of mass w.r.t time must vanish i.e.

$$\frac{D}{Dt} \int_V \rho(\vec{x}, t) dV = 0. \quad (2.7)$$

Applying Reynold's Transport Theorem, we have

$$\frac{D}{Dt} \int_V \rho(\vec{x}, t) dV = \int_V \left\{ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) \right\} dV = 0. \quad (2.8)$$

Since we are working within the confines of the continuum hypothesis and since this equation must hold true for arbitrarily small domains, we can infer that

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0. \quad (2.9)$$

which is the continuity equation for compressible flows.

Expanding the divergence term in Eq. (2.9) yields

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \vec{u} = 0. \quad (2.10)$$

where  $\frac{D}{Dt} = \frac{\partial}{\partial t} + (\vec{u} \cdot \nabla)$

In the case where the density remains constant, Eq.(2.9) simplifies to

$$\nabla \cdot \vec{u} = 0 \quad (2.11)$$

which is the continuity equation for incompressible flows.

## 2.3 Conservation of Momentum

The Conservation of Linear Momentum Principle states that the rate of change of momentum w.r.t time is equal to the sum of the acting forces. This is given mathematically by

$$\frac{D}{Dt} \int_V \rho(\vec{x}, t) \vec{u} dV = \sum \text{acting forces} \quad (2.12)$$

The forces of interest are body forces such as gravity, Coriolis force etc. and surface forces such as pressure and internal friction(viscosity).

Body forces will be expressed as

$$\int_V \rho(\vec{x}, t) g(\vec{x}, t) dV \quad (2.13)$$

and surface forces will be expressed as

$$\int_{\partial V} \sigma(\vec{x}, t) \vec{n} ds \quad (2.14)$$

where  $\sigma$  is the stress tensor,  $\vec{n}$  is the outward unit normal, and  $\partial V$  is the boundary of the domain  $V$

Recasting Newton's 2nd law with the aforementioned information in mind, we have

$$\frac{D}{Dt} \int_V \rho(\vec{x}, t) \vec{u} dV = \int_V \rho(\vec{x}, t) g(\vec{x}, t) dV + \int_{\partial V} \sigma(\vec{x}, t) \vec{n} ds \quad (2.15)$$

Applying Reynold's Transport Theorem to the term on the left yields

$$\int_V \left\{ \frac{\partial \rho}{\partial t} \vec{u} + \nabla \cdot \rho \vec{u} \vec{u} \right\} dV = \int_V \rho(\vec{x}, t) g(\vec{x}, t) dV + \int_{\partial V} \sigma(\vec{x}, t) \vec{n} ds \quad (2.16)$$

After applying the Divergence Theorem to the last term and regrouping the equation we arrive at

$$\frac{\partial \rho}{\partial t} \vec{u} + (\vec{u} \cdot \nabla) \rho \vec{u} + \rho \vec{u} \nabla \cdot \vec{u} - \rho \vec{g} - \nabla \cdot \sigma = 0 \quad (2.17)$$

The stress tensor  $\sigma$  for viscous fluids is defined as

$$\sigma := -pI + \tau := -p + \lambda \text{div}(\vec{u})I + 2\mu\delta. \quad (2.18)$$

Using this stress tensor in Eq. (2.17), yields

$$\frac{\partial \rho}{\partial t} \vec{u} + (\vec{u} \cdot \nabla) \rho \vec{u} + \rho \vec{u} \nabla \cdot \vec{u} + \text{grad} p = (\mu + \lambda) \text{grad}(\text{div}(\vec{u})) + \mu \Delta \vec{u} + \rho \vec{g} \quad (2.19)$$

If the fluid being studied is incompressible (i.e  $\rho = \rho_\infty = \text{constant}$ ), Eq. (2.19)

reduces to

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} + \frac{1}{\rho_\infty} \text{grad} p = \frac{\mu}{\rho_\infty} \Delta \vec{u} + \vec{g} \quad (2.20)$$

Equation (2.11) together with Eq.(2.20) comprise what is today known as the Navier-Stokes Equations for incompressible flows.



## 2.4 Dimensional Analysis

Performing experiments for flow applications can be very expensive and time consuming. Take for example studying the flow of air around automobiles. One would need to build the automobile as well as reproduce the conditions under which they operate. As we know, this is more easily said than done. These kinds of applications motivated the need for prototypes and models. The first step in this process is to nondimensionalize the governing equations. This is done by selecting reference quantities for each of the variable present in the equation. Those variable are  $\vec{u}, \vec{x}, p$ , and  $t$  and the associated reference quantities are  $V, L, \rho_\infty V^2$ , and  $\frac{L}{V}$ . A quick check will show that these variables and their associated reference quantities are dimensionally homogeneous. Because of this, the dimensionless quantities can be defined as

$$\vec{u}^* := \frac{\vec{u}}{V}, \vec{x}^* := \frac{\vec{x}}{L}, p^* := \frac{p}{\rho_\infty V^2}, t^* := \frac{Vt}{L} \quad (2.21)$$

Recasting equation (15) with these new variable yields,

$$\frac{V^2}{L} \frac{\partial \vec{u}^*}{\partial t^*} + \frac{V^2}{L} (\vec{u}^* \cdot \text{grad}^*) \vec{u}^* + \frac{V^2}{L} \text{grad}^* p^* = \frac{V\mu}{L^2 \rho_\infty} \Delta^* \vec{u}^* + \vec{g} \quad (2.22)$$

Multiplying through by  $\frac{L}{V^2}$  yields,

$$\frac{\partial \vec{u}^*}{\partial t^*} + (\vec{u}^* \cdot \text{grad}^*) \vec{u}^* + \text{grad}^* p^* = \frac{\mu}{L\rho_\infty V} \Delta^* \vec{u}^* + \frac{L}{V^2} \vec{g} \quad (2.23)$$

With the introduction of the dimensionless body force

$$\vec{g}^* := \frac{L}{V^2} \vec{g} = \frac{1}{Fr^2} \frac{\vec{g}}{\|\vec{g}\|} \quad (2.24)$$

the equation becomes

$$\frac{\partial \vec{u}^*}{\partial t^*} + (\vec{u}^* \cdot \text{grad}^*) \vec{u}^* + \text{grad}^* p^* = \frac{\mu}{L\rho_\infty V} \Delta^* \vec{u}^* + \vec{g}^* \quad (2.25)$$

The coefficient of the Laplacian gives the reciprocal of the Reynolds number and the last term involves the reciprocal of the square of the Froude number. Recasting Eq. (2.25) with this in mind and dropping the \* for convenience yields,

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \text{grad}) \vec{u} + \text{grad} p = \frac{1}{Re} \Delta \vec{u} + \vec{g} \quad (2.26)$$

Doing this for the continuity equation will leave it unchanged. It is worth mentioning that the dimensionless quantities that arise in this equation depend heavily on the the reference quantities which usually depend on what is of interest in the flow being analyzed [3].

## 2.5 Geometries and Boundary Conditions

As stated previously, the Navier-Stokes Equations are the governing equations for fluid flow. Having this in mind, a question that might arise quite naturally is how does one obtain different flow fields as it relates to various applications. The answer to this question resides in the fact that a specific geometry and various boundary conditions must be supplied when using the Navier-Stokes Equations for a particular application. The boundary conditions that are of particular interest are the Dirichlet boundary condition(no-slip boundary condition), Neumann boundary conditions, mixed boundary conditions, and periodic boundary conditions. Let  $\varphi_n$  denote the component of velocity orthogonal to the boundary in the exterior normal direction,  $\varphi_t$  the component of velocity parallel to the boundary in the tangential direction, and  $\frac{\partial \varphi_n}{\partial n}$  and  $\frac{\partial \varphi_t}{\partial n}$  be their respective derivatives in the normal direction.

### 2.5.1 Dirichlet Boundary Condition

Dirichlet boundary conditions specify the values a solution must take on the boundary of the domain. In the case of fluid flow problems, the no-slip con-

ditions, which are homogeneous Dirichlet boundary conditions, are often enforced on the boundaries of the given geometry. The no-slip condition says that no fluid penetrates the boundary and the fluid is at rest there. Mathematically, this is given by

$$\varphi_n = 0, \quad \varphi_t = 0 \quad (2.27)$$

These are to be understood irrespective of the boundary being investigated.

### 2.5.2 Neumann Boundary Condition

Neumann boundary conditions specify the values that the directional derivative must satisfy along the outward normal to the boundary of the domain. A situation where this occurs is when outflow conditions, which are homogeneous Neumann boundary conditions, are supplied. They state that neither velocity component changes in the direction normal to the boundary. Mathematically, this is given by

$$\frac{\partial \varphi_n}{\partial n} = 0, \quad \frac{\partial \varphi_t}{\partial n} = 0. \quad (2.28)$$

The partial derivatives are taken with respect to different axes depending on the orientation of the boundaries.

### 2.5.3 Mixed Boundary Condition

Mixed boundary conditions are combinations of Dirichlet and Neumann boundary conditions. A situation where this occurs is when free-slip conditions, which are homogeneous mixed boundary conditions, are supplied. They state that no fluid penetrates the boundary and there are no frictional losses at the boundary. Mathematically, this is given by

$$\varphi_n = 0, \quad \frac{\partial \varphi_t}{\partial n} = 0 \quad (2.29)$$

### 2.5.4 Periodic Boundary Condition

Periodic boundary conditions occur when the flow problem is periodic over one or more of the coordinate directions. A situation where this occurs is flow over an undulating surface. Since the values on the boundaries are repeated over some period  $a$ , the computations can be restricted to one period interval.

## 2.6 Conclusion

In this chapter, we presented the derivation of the Navier-Stokes equations from the conservation of mass and linear momentum. In order to do this, we use Reynold's Transport Theorem. Following the derivation, dimensional analysis of the equations is presented and boundary conditions are discussed so that we have a well-posed problem to work with. In the next chapter, we move from the continuous problem to the discrete problem.

## Chapter 3

# Discretization of the Navier-Stokes Equations

In this chapter, we discuss the finite-differencing scheme for the discretization of the Navier-Stokes equations as well as describe in detail Chorin's Projection Method for solving the resulting Poisson equation for pressure.

## 3.1 Finite Difference Approximations

### 3.1.1 First Order Finite Difference Approximations

In order to work efficiently with the series that follow, we introduce the concept of the Big-O notation.

**Definition 3.1** *Let  $f(x)$  and  $g(x)$  be two functions defined on a subset of real numbers. We say that  $f(x)$  is **Big-O** of  $g(x)$  if there exist positive real numbers  $C$  and  $x_0$  such that*

$$|f(x)| \leq C |g(x)| \quad \forall x > x_0$$

*This can also be formulated to study the behavior of  $f(x)$  near some real number  $a$  as follows: Let  $f(x)$  and  $g(x)$  be two functions defined on a subset of real numbers. We say that  $f(x)$  is **Big-O** of  $g(x)$  if there exist positive real numbers  $C$  and  $\delta$  such that*

$$|f(x)| \leq C |g(x)| \quad \text{for } |x - a| < \delta$$



In order to perform the discretization of the governing equation for fluid motion, one must be able to approximate the continuous differential operator at a finite number of grid points. Consider a function  $u(x, t)$  with  $x \in [0, L]$  and  $t \in [0, T]$ . A discretization of the function  $u$  is obtained by considering only the values  $u_i^j$  at a finite number of grid points  $(x_i, t_j)$ . Applying Taylors series expansion to  $u(x_i, t_j + \delta t)$  gives

$$u(x_i, t_j + \delta t) = u(x_i, t_j) + u_t(x_i, t_j)\delta t + u_{tt}(x_i, t_j)\frac{(\delta t)^2}{2} + u_{ttt}(x_i, t_j)\frac{(\delta t)^3}{3!} + O((\delta t)^4) \quad (3.1)$$

The following convention will be used in an effort to reduce the use of notation throughout this work.

$$u_i^j = u(x_i, t_j) = u(i\delta x, j\delta t) \quad i = 0, 1, \dots, m; \quad j = 0, 1, \dots, n. \quad (3.2)$$

Recasting the above equation with this in mind yields

$$u_i^{j+1} = u_i^j + (u_t)_i^j \delta t + (u_{tt})_i^j \frac{(\delta t)^2}{2} + (u_{ttt})_i^j \frac{(\delta t)^3}{3!} + O((\delta t)^4). \quad (3.3)$$

Truncating this equation at the second order term and solving for  $(u_t)_i^j$  yields

$$(u_t)_i^j = \frac{u_i^{j+1} - u_i^j}{\delta t} + O(\delta t). \quad (3.4)$$

Therefore

$$(u_t)_i^j \approx \frac{u_i^{j+1} - u_i^j}{\delta t} \quad (3.5)$$

This equation is known as the two point forward finite difference approximation of  $\frac{\partial u}{\partial t}$ .

Applying Taylors series expansion to  $u_i^{j-1}$  gives

$$u_i^{j-1} = u_i^j - (u_t)_i^j \delta t + (u_{tt})_i^j \frac{(\delta t)^2}{2} - (u_{ttt})_i^j \frac{(\delta t)^3}{3!} + O((\delta t)^4) \quad (3.6)$$

Truncating this equation at the second order term and solving for  $(u_t)_i^j$  yields

$$(u_t)_i^j = \frac{u_i^j - u_i^{j-1}}{\delta t} + O(\delta t). \quad (3.7)$$

Therefore

$$(u_t)_i^j \approx \frac{u_i^j - u_i^{j-1}}{\delta t} \quad (3.8)$$

This equation is known as the two point backward finite difference approximation of  $\frac{\partial u}{\partial t}$ .

Doing the difference of equations (ffd) and (bfd) yield

$$u_i^{j+1} - u_i^{j-1} = 2(u_t)_i^j \delta t + O((\delta t)^3). \quad (3.9)$$

Solving for  $(u_t)_i^j$  yields

$$(u_t)_i^j = \frac{u_i^{j+1} - u_i^{j-1}}{2\delta t} + O((\delta t)^2). \quad (3.10)$$

Therefore

$$(u_t)_i^j \approx \frac{u_i^{j+1} - u_i^{j-1}}{2\delta t} \quad (3.11)$$

This equation is known as the two point central difference approximation of  $\frac{\partial u}{\partial t}$ .

The forward and backward finite difference approximation of  $\frac{\partial u}{\partial t}$  have an error of order  $\delta t$  where as the central finite difference approximation of  $\frac{\partial u}{\partial t}$  has an error of order  $(\delta t)^2$  which indicates that for most applications, the central finite difference approximation will yield more accurate results. All of the finite difference approximations for the spatial derivatives can be found in an analogous way to the difference approximations found for the time derivative.

### 3.1.2 Higher Order Finite Difference Approximations

Following a similar procedure to that used in the previous section, higher order finite difference approximations can be derived but only the second order central finite difference approximations will be investigated here.

To accomplish this, the Taylor Series Expansion is again found for  $u_i^{j+1}$  and  $u_i^{j-1}$ .

$$u_i^{j-1} = u_i^j - (u_t)_i^j \delta t + (u_{tt})_i^j \frac{(\delta t)^2}{2} - (u_{ttt})_i^j \frac{(\delta t)^3}{3!} + O((\delta t)^4) \quad (3.12)$$

$$u_i^{j+1} = u_i^j + (u_t)_i^j \delta t + (u_{tt})_i^j \frac{(\delta t)^2}{2} + (u_{ttt})_i^j \frac{(\delta t)^3}{3!} + O((\delta t)^4). \quad (3.13)$$

Taking the sum of these expansions yields

$$u_i^{j-1} + u_i^{j+1} = 2u_i^j + (u_{tt})_i^j (\delta t)^2 + O((\delta t)^4) \quad (3.14)$$

Therefore,

$$(u_{tt})_i^j \approx \frac{u_i^{j-1} - 2u_i^j + u_i^{j+1}}{(\delta t)^2} \quad (3.15)$$

This equation is known as the three point central difference approximation of  $\frac{\partial^2 u}{\partial t^2}$ .

The second order forward and backward finite difference approximations can be found by composing any of the first order approximations.

When dealing with Navier-Stokes equations, using a purely finite-difference discretization would lead to stability issues. We introduce the Donor-Cell discretization scheme in order to circumvent this stability issue.

## 3.2 The Donor-Cell Discretization Scheme

We define the donor-cell discretization [2] of some convective term  $\frac{d(ku)}{dx}$  by

$$\left[ \frac{d(ku)}{dx} \right]_i^{dc} := \frac{k_r u_r - k_l u_l}{\delta x} \quad (3.16)$$

where  $u_r$  and  $u_l$  are chosen based on the sign of  $k_r$  and  $k_l$ , i.e.

$$u_r := \begin{cases} u_i & k_r > 0 \\ u_{i+1} & k_r < 0 \end{cases} \quad u_l := \begin{cases} u_{i-1} & k_l > 0 \\ u_i & k_l < 0 \end{cases}$$

In order to avoid these case distinction, the donor-cell discretization can written as

$$\left[ \frac{d(ku)}{dx} \right]_i^{dc} := \frac{1}{2\delta x} (k_r(u_i + u_{i+1}) - k_l(u_{i-1} + u_i) + |k_r|(u_i - u_{i+1}) - |k_l|(u_{i-1} - u_i)) \quad (3.17)$$

The need for this scheme will be explained in the sections covering the discretization of the Navier-Stokes equations.

### 3.3 Discretization of the Heat Equation

In this subsection, we introduce the finite difference approximation of the one-dimensional heat equation without sources on a finite interval  $0 < x < L$ . Handling of the initial and boundary conditions will be ignored here and will be dealt with in detail for the Navier-Stokes Equations. The Heat Equation as specified is given by

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2} \\ u(0, t) = 0 \\ u(L, t) = 0 \\ u(x, 0) = f(x) \end{array} \right.$$

In order to discretize this equation, we replace the time derivative with its forward finite difference approximation and the space derivative with its central finite difference approximation yielding

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = k \frac{u_{i-1}^j - 2u_i^j + u_{i+1}^j}{(\Delta x)^2}. \quad (3.18)$$

The forward finite difference approximation was used for the time derivative because of the initial condition supplied and the central finite difference approxima-

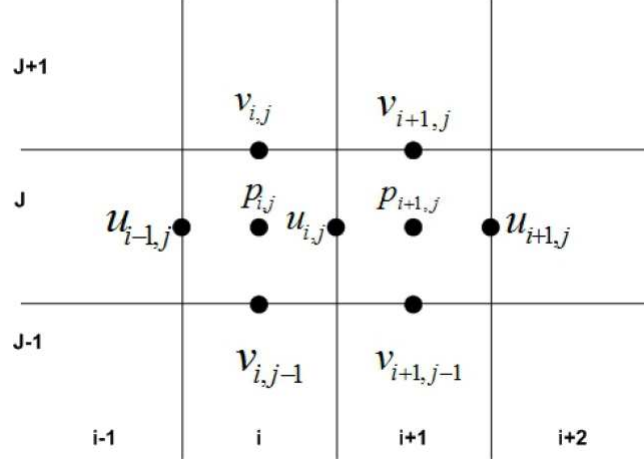
tion was used for the space derivative because of the boundary conditions supplied. Since the truncated terms go to zero as  $\Delta t$  and  $\Delta x$  go to zero, the numerical approximate is said to be **consistent** with the given PDE in its continuous form. The analysis for Navier-Stokes equations are now investigated.

### 3.4 Discretization of the Spatial Derivative

We use a staggered grid in order to discretize the Navier-Stokes Equations [2]. In this approach, the horizontal velocities  $u$  are located at the midpoint of the vertical cell edges, the vertical velocities  $v$  are located at the midpoint of the horizontal cell edges, and the pressure  $p$  is located at the cell centers. Using the staggered grid approach eliminates the pressure oscillations that would occur with a grid in which  $u, v$ , and  $p$  are all located at the same grid point. In order to discretize the continuity equation, we replace the spatial derivatives  $\frac{\partial u}{\partial x}$  and  $\frac{\partial v}{\partial y}$  by centered differences with half the mesh width. That is,

$$\left[ \frac{\partial u}{\partial x} \right]_i^j := \frac{u_i^j - u_{i-1}^j}{\delta x} \qquad \left[ \frac{\partial v}{\partial y} \right]_i^j := \frac{v_i^j - v_i^{j-1}}{\delta y} \qquad (3.19)$$

Figure 3.1: Staggered Grid



The terms  $\frac{\partial^2 u}{\partial x^2}$ ,  $\frac{\partial^2 u}{\partial y^2}$ ,  $\frac{\partial^2 v}{\partial x^2}$ , and  $\frac{\partial^2 v}{\partial y^2}$  in the momentum equation are replaced with the second order centered difference approximations as discussed earlier in the chapter. These terms form the so called diffusive terms of the momentum equations. The remaining terms present small complications due to the form of the grid spacing that was chosen. For the terms,  $\frac{\partial uv}{\partial y}$  and  $\frac{\partial u^2}{\partial x}$  we use the averages of  $\mathbf{u}, \mathbf{v}$  and  $\mathbf{u}$  respectively to obtain suitable values of the product lying in the two vertical directions. That is,

$$\left[ \frac{\partial(uv)}{\partial y} \right]_i^j := \frac{1}{\delta y} \left( \frac{(v_i^j + v_{i+1}^j)(u_i^j + u_{i+1}^{j+1})}{2} - \frac{(v_i^{j-1} + v_{i+1}^{j-1})(u_i^{j-1} + u_{i+1}^j)}{2} \right) \quad (3.20)$$

and



$$\left[ \frac{\partial(u^2)}{\partial x} \right]_i^j := \frac{1}{\delta x} \left( \left( \frac{u_i^j + u_{i+1}^j}{2} \right)^2 - \left( \frac{u_{i-1}^j + u_i^j}{2} \right)^2 \right) \quad (3.21)$$

The terms  $\frac{\partial(uv)}{\partial x}$  and  $\frac{\partial(u^2)}{\partial x}$  are treated in an analogous fashion. Since at high Reynolds numbers the convective terms in our equation become dominant and may present problems with stability, a blend of the central differences and the donor-cell discretization is used. This leads to the following discretization.

$$\begin{aligned} \left[ \frac{\partial(u^2)}{\partial x} \right]_i^j &:= \frac{1}{\delta x} \left( \left( \frac{u_i^j + u_{i+1}^j}{2} \right)^2 - \left( \frac{u_{i-1}^j + u_i^j}{2} \right)^2 \right) \\ &+ \gamma \frac{1}{\delta x} \left( \frac{|u_i^j + u_{i+1}^j|}{2} \frac{(u_i^j - u_{i+1}^j)}{2} - \frac{|u_{i-1}^j + u_i^j|}{2} \frac{(u_{i-1}^j - u_i^j)}{2} \right) \end{aligned}$$

$$\begin{aligned} \left[ \frac{\partial(uv)}{\partial y} \right]_i^j &:= \frac{1}{\delta y} \left( \frac{(v_i^j + v_{i+1}^j)}{2} \frac{(u_i^j + u_{i+1}^j)}{2} - \frac{(v_i^{j-1} + v_{i+1}^{j-1})}{2} \frac{(u_i^{j-1} + u_{i+1}^{j-1})}{2} \right) \\ &+ \gamma \frac{1}{\delta y} \left( \frac{|v_i^j + v_{i+1}^j|}{2} \frac{(u_i^j - u_{i+1}^j)}{2} - \frac{|v_i^{j-1} + v_{i+1}^{j-1}|}{2} \frac{(u_i^{j-1} - u_{i+1}^{j-1})}{2} \right) \end{aligned}$$

$$\left[ \frac{\partial^2 u}{\partial x^2} \right]_i^j := \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\delta x)^2}$$

$$\left[ \frac{\partial^2 u}{\partial y^2} \right]_i^j := \frac{u_i^{j+1} - 2u_i^j + u_i^{j-1}}{(\delta y)^2}$$

$$\left[ \frac{\partial p}{\partial x} \right]_i^j := \frac{p_{i+1}^j - p_i^j}{(\delta x)}$$

$$\begin{aligned} \left[ \frac{\partial(v^2)}{\partial y} \right]_i^j &:= \frac{1}{\delta y} \left( \left( \frac{v_i^j + v_i^{j+1}}{2} \right)^2 - \left( \frac{v_i^{j-1} + v_i^j}{2} \right)^2 \right) \\ &+ \gamma \frac{1}{\delta y} \left( \frac{|v_i^j + v_i^{j+1}|}{2} \frac{(v_i^j - u_i^{j+1})}{2} - \frac{|v_i^{j-1} + v_i^j|}{2} \frac{(v_i^{j-1} - v_i^j)}{2} \right) \end{aligned}$$

$$\begin{aligned} \left[ \frac{\partial(uv)}{\partial x} \right]_i^j &:= \frac{1}{\delta x} \left( \frac{(u_i^j + u_i^{j+1})}{2} \frac{(v_i^j + v_{i+1}^j)}{2} - \frac{(u_{i-1}^j + u_{i-1}^{j+1})}{2} \frac{(v_{i-1}^j + v_i^j)}{2} \right) \\ &+ \gamma \frac{1}{\delta x} \left( \frac{|u_i^j + u_i^{j+1}|}{2} \frac{(v_i^j - v_{i+1}^j)}{2} - \frac{|u_{i-1}^j + u_{i-1}^{j+1}|}{2} \frac{(v_{i-1}^j - v_i^j)}{2} \right) \end{aligned}$$

$$\left[ \frac{\partial^2 v}{\partial x^2} \right]_i^j := \frac{v_{i+1}^j - 2v_i^j + v_{i-1}^j}{(\delta x)^2}$$

$$\left[ \frac{\partial^2 v}{\partial y^2} \right]_i^j := \frac{v_i^{j+1} - 2v_i^j + v_i^{j-1}}{(\delta y)^2}$$

$$\left[ \frac{\partial p}{\partial y} \right]_i^j := \frac{p_i^{j+1} - p_i^j}{(\delta y)}$$

The parameter  $\gamma$  lies between 0 and 1 and should be chosen such that

$$\gamma \geq \max_{i,j} \left( \left| \frac{u_{i,j} \delta t}{\delta x} \right|, \left| \frac{v_{i,j} \delta t}{\delta y} \right| \right)$$

Appropriately discretized boundary conditions must also be supplied depending on the nature of the problem being studied.

### 3.5 Discretization of the Time Derivative

In order to discretize the time derivatives occurring in the momentum equations, first order finite differences are used [2]. Doing this yields

$$\left[ \frac{\partial u}{\partial t} \right]^{(n+1)} := \frac{u^{(n+1)} - u^{(n)}}{\delta t}, \quad \left[ \frac{\partial v}{\partial t} \right]^{(n+1)} := \frac{v^{(n+1)} - v^{(n)}}{\delta t} \quad (3.22)$$

### 3.5.1 The Time-Stepping Algorithm

The time-stepping algorithm begins by discretizing the time derivatives appearing in the momentum equation. As a result, we have

$$u^{(n+1)} = u^{(n)} + \delta t \left[ \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial(u^2)}{\partial x} - \frac{\partial(uv)}{\partial y} + g_x - \frac{\partial p}{\partial x} \right] \quad (3.23)$$

$$v^{(n+1)} = v^{(n)} + \delta t \left[ \frac{1}{Re} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial(uv)}{\partial x} - \frac{\partial(v^2)}{\partial y} + g_y - \frac{\partial p}{\partial y} \right] \quad (3.24)$$

We introduce the abbreviations

$$F := u^{(n)} + \delta t \left[ \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial(u^2)}{\partial x} - \frac{\partial(uv)}{\partial y} + g_x - \frac{\partial p}{\partial x} \right] \quad (3.25)$$

$$G := v^{(n)} + \delta t \left[ \frac{1}{Re} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial(uv)}{\partial x} - \frac{\partial(v^2)}{\partial y} + g_y - \frac{\partial p}{\partial y} \right] \quad (3.26)$$

and associate to them the time level  $n$  to arrive at

$$u^{(n+1)} = F^{(n)} - \delta t \frac{\partial p}{\partial x} \quad v^{(n+1)} = G^{(n)} - \delta t \frac{\partial p}{\partial y} \quad (3.27)$$

We associate the time level  $(n + 1)$  to the pressure term yielding

$$u^{(n+1)} = F^{(n)} - \delta t \frac{\partial p^{(n+1)}}{\partial x} \quad v^{(n+1)} = G^{(n)} - \delta t \frac{\partial p^{(n+1)}}{\partial y} \quad (3.28)$$

Upon substituting this equation into the continuity equation, we obtain,

$$\frac{\partial u^{(n+1)}}{\partial x} + \frac{\partial v^{(n+1)}}{\partial y} = \frac{\partial F^{(n)}}{\partial x} - \delta t \frac{\partial^2 p^{(n+1)}}{\partial x^2} + \frac{\partial G^{(n)}}{\partial y} - \delta t \frac{\partial^2 p^{(n+1)}}{\partial y^2} = 0 \quad (3.29)$$

Therefore,

$$\frac{\partial^2 p^{(n+1)}}{\partial x^2} + \frac{\partial^2 p^{(n+1)}}{\partial y^2} = \frac{1}{\delta t} \left( \frac{\partial F^{(n)}}{\partial x} + \frac{\partial G^{(n)}}{\partial y} \right) \quad (3.30)$$

which is a Poisson equation for pressure [2].

In order to solve this resulting Poisson equation, boundary values for the pressure are required. These result from multiplying the time-discrete momentum equation

with the exterior unit normal  $\vec{n} := (n_1, n_2)^T$  on the boundary yielding,

$$\begin{aligned} \text{grad} p^{(n+1)} \cdot \vec{n} &= \frac{\partial p^{(n+1)}}{\partial x} n_1 + \frac{\partial p^{(n+1)}}{\partial y} n_2 \\ &= -\frac{1}{\delta t} \left( (u^{(n+1)} - F^{(n)}) n_1 + (v^{(n+1)} - G^{(n)}) n_2 \right) \end{aligned}$$

Discretization of the Poisson equation will result in a linear system of equation. Because of the computational cost that arises using elementary methods for solving this system such as Gaussian Elimination, a technique known as Successive Over Relaxation(SOR) [4] is used . Successive Over Relaxation is a generalization of the more familiar Gauss-Seidel and Jacobi Method.

The above approach is known as Chorin projection method and is attributed to Alexandre Joel Chorin [1].

In order to get the fully discretized momentum equations, we take the time-discretized momentum equations and associate to the spatial derivatives their corresponding spatial discretizations. Doing so yields,

$$\bar{\bar{u}}_i^j = \bar{F}_i^j - \frac{\delta t}{\delta x} (\bar{\bar{p}}_{i+1}^j - \bar{\bar{p}}_i^j) \quad (3.31)$$

$$\bar{\bar{v}}_i^j = \bar{G}_i^j - \frac{\delta t}{\delta y} (\bar{\bar{p}}_i^{j+1} - \bar{\bar{p}}_i^j) \quad (3.32)$$

where  $\bar{\bar{u}}$  stands for  $u$  with associated time level  $(n+1)$ , and  $\bar{u}$  stands for  $u$  with associated time level  $(n)$ .

In order to maintain stability of the presented algorithm, the time step  $\delta t$  is chosen such that the following conditions hold.

$$\frac{2\delta t}{Re} < \left( \frac{1}{\delta x^2} + \frac{1}{\delta y^2} \right)^{-1}, \quad |u_{max}| \delta t < \delta x, \quad |v_{max}| \delta t < \delta y \quad (3.33)$$

The latter two of these conditions are known as the famous Courant-Friedrichs-Lewy(CFL) conditions. They state that a fluid particle cannot, in any given time step, travel a distance greater than the grid spacing. Choosing  $\delta t$  to be the minimal for these conditions guarantees stability of the algorithm, i.e.

$$\delta t := \tau \min \left( \frac{Re}{2} \left( \frac{1}{\delta x^2} + \frac{1}{\delta y^2} \right)^{-1}, \frac{\delta x}{|u_{max}|}, \frac{\delta y}{|v_{max}|} \right) \quad (3.34)$$

where  $\tau \in [0, 1]$  is a safety factor.

## 3.6 Conclusion

In this chapter, we presented the discretization of the Navier-Stokes equations using the finite-differencing technique. Various finite-difference approximations of the derivatives appearing in our equation were derived as well as the Donor-Cell discretization technique for handling stability issues that arise when dealing with the discretization of the convective terms of the Navier-Stokes equations. From this point, Chorin Projection method was presented to enforce the incompressibility condition. In the next chapter, we present the functions needed to implement the algorithm discussed in this chapter.



## Chapter 4

# Implementation of the Navier-Stokes Equations Solver

In this chapter, we discuss the functions needed to implement the algorithm for finding the numerical solutions to Navier-Stokes equations. All these functions were implemented in C by Griebel *et al.*[2]. The program is available as an open source software called Nast2d.

## 4.1 The Program

The following is an explanation of the input parameters and functions used in the aforementioned open source code.

- `inputfile` is the inputfile name
- `problem` is the specific predefined problem types
- `xlength` is the length of the computational domain in the x direction
- `ylength` is the length of the computational domain in the y direction
- `imax` is the number of subinterval of `xlength`
- `jmax` is the number of subinterval of `ylength`
- `delx` is the spatial step size in the x direction
- `dely` is the spatial step size in the y direction
- `delt` is the time step size
- $t_{end}$  is the final time level
- `tau` is a safety factor used when choosing the time step

- `itermax` is the maximum number of iterations
- `eps` is the stopping tolerance for the pressure iteration
- `omg` successive overrelaxation parameter
- `gamma` is the parameter chosen to maintain stability for convective terms
- `Re` is the Reynold's number
- `GX` is the body force in the x direction
- `GY` is the body force in the y direction
- `UI` is the initial velocity in the horizontal direction
- `VI` is the initial velocity in the vertical direction
- `PI` is the initial pressure
- `wW` is the western boundary
- `wE` is the eastern boundary
- `wN` is the northern boundary
- `wS` is the southern boundary

Using the described variables, the following functions are used to implement the algorithm.

- **ReadParameter**( inputfile, problem, xlength, ylength, imax, jmax, delx, dely, delt,  $t_{end}$ , tau, itermax, eps, omg, gamma, Re, GX, GY, UI, VI, PI, wW, wE, wN, wS )

This function is tasked with reading all of the input parameters required for the program.

- **InitUVP**( U, V, P, imax, jmax, UI, VI, PI)

This function is tasked with initializing the arrays  $U$ ,  $V$ , and  $P$  with their corresponding initial values  $UI$ ,  $VI$ , and  $PI$ .

- **CompDelt**( delt, imax, jmax, delx, dely, U, V, Re, tau)

This function is tasked with calculating the stepsize delt for the next time step according to equation(). If  $\tau \leq 0$ , the stepsize read in from ReadParameter is to be used.

- **SetBCond**( U, V, imax, jmax, wW, wE, wN, wS)

This function is tasked with setting the boundary data parameters wW, wE, wN, wS for the arrays U and V.

- **SetSPECBcond**( U, V, imax, jamx, wW, wE, wN, wS)

This function is tasked with specifying problem specific boundary conditions.

- **CompFG**( U, V, F, G, imax, jamx, delt, delx, dely, GX, GY, gamma, Re)

This function is tasked with the computation of  $F$  and  $G$  according to equation().

- **CompRHS**( F, G, RHS, imax, jmax, delt, delx, dely)

This function is tasked with the computation of the right-hand side of the pressure equation.

- **Poisson**( P, RHS, imax, jmax, delx, dely, eps, itermax, omg, res)

This function is tasked with performing the successive over relaxation iteration for the pressure Poisson equation and is terminated once the residual norm reaches its tolerance eps, or once the maximum number of iterations(itermax) is reached.

- **AdapUV**( U, V, F, G, P, imax, jmax, delt, delx, dely)

This function is tasked with calculating the new velocities according to Eq. (3.27).

## 4.2 Conclusion

In this chapter, we presented the functions needed for implementation of the aforementioned algorithm. In the next chapter, techniques to visualize the numerical solutions obtained are presented.

# Chapter 5

## Visualization Techniques

Being able to visualize the results obtained from the implemented algorithm discussed in the previous chapter is a very important part of the analysis process. When dealing with fluid flow problems, having visual data can go a long way in interpreting numerical results. In this chapter, we discuss some visualization techniques that can be used when working with the Navier-Stokes equations.

## 5.1 Vector Field Representation

A 2-D vector field can be used to visualize various flow field problems. It can be visualized as arrows, with a given magnitude and direction, attached to each point in the plane. The vectors are drawn by taking the corresponding vector values for the horizontal velocity  $u$  and the vertical velocity  $v$  and finding their geometric sum.

## 5.2 Streamlines

**Definition 5.1** A ***streamline*** is a line whose tangent at each point is in the direction of the velocity vector at that point. These plots appear as contour graphs once visualized.

## 5.3 Conclusion

In this chapter, we investigated two ways to visualize the numerical solutions of the Navier-Stokes equations. Open source Matlab programs were used for the



visualization. Other commercial as well as non-commercial visualization softwares are available for use depending on the ability of the user and results desired. In the chapter to follow, we use these visualization techniques to present different fluid flow problems.

# Chapter 6

## Examination of Classical Flows

### 6.1 Flow Over a Flat Plate

Using a similarity transform, we look for a stream function  $\psi(x, y)$  in the form

$$\psi(x, y) = \sqrt{(\nu U_0 x)} f(\eta) \quad \eta = y \sqrt{\frac{U_0}{\nu x}}.$$

After simplification, we arrive at

$$f'''(\eta) + \frac{1}{2}f(\eta)f''(\eta) = 0$$

which is Blasius equation. It is subject to the boundary conditions

$$y = 0 : V_x = 0 \quad \eta = 0, f'(0) = 0$$

$$y = 0 : V_y = 0 \quad \eta = 0, f(0) = 0$$

$$y = \infty : V_x = U_0 \quad \eta = 0, f'(\infty) = 1.$$

The friction on the lower plate is proportional to  $f''(0) \approx 0.332$ . In order to confirm this result using Nast2d, we simulated a flow over a flat plate as follows. We solve the 2-D Navier-Stokes equation until we reach steady state. We used the following physical parameters. The length of the plate was chosen to be 30 and the inflow velocity is  $U_0 = 1$ . The Reynolds number for the flow was taken to be 500,000. A plate moving with a velocity equal to the inflow velocity was placed above the flow region. We found that there is a small region close to the lower boundary in which the effect of viscosity is pronounced. Far from the lower boundary, the flow is almost uniform. We have provided the velocity profile in Figure (6.1) and we have given few numerical values from our computation in Table(6.1). Steady state was reached at approximately  $t = 2000$ . We obtained the approximate value  $f''(0) \approx .2645$  by using the similarity profile for  $x = 15$ . This similarity profile is given in Figure(6.1). There is a relative error of approximately 20% with the theoretical value. Such error may be attributed to several causes

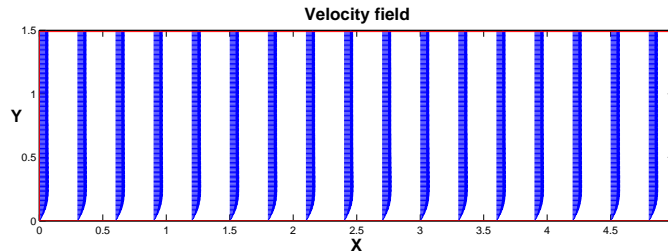


Figure 6.1: Flow Between Two Parallel Plates at  $Re = 500,000$

including the fact that steady state was reached approximately in our computations or from rounding errors.

## 6.2 Flow Between Two Parallel Plates

The velocity profile of a fluid flow driven by an inflow between two parallel plates is given in figure(6.2) for  $Re = 500$ . We employ the same inflow velocity as before in all our simulations. In the case where  $Re = 500,000$  the effect of viscosity is pronounced in a small layer close to the boundary. This small region is known as

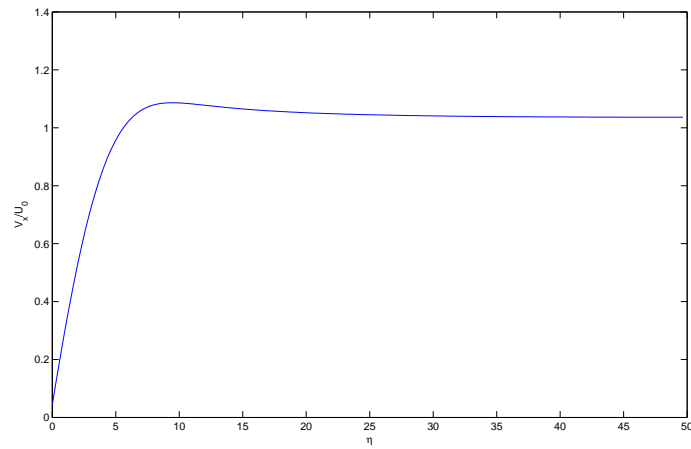


Figure 6.2: Flow Between Two Parallel Plates at  $Re = 500,000$

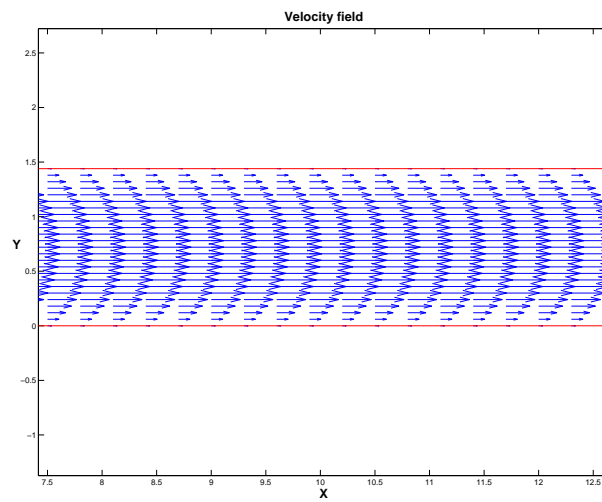


Figure 6.3: Flow Between Two Parallel Plates at  $Re = 500$

Table 6.1: Velocity in the  $x$ -direction vs. similarity variable at  $x = 15$ 

$V_x/U_0$	$\eta$
0.044572	0
0.13274	0.333333333
0.218923	0.666666667
0.302261	1
0.382128	1.333333333
0.458035	1.666666667
0.529589	2
0.596505	2.333333333
0.658584	2.666666667
0.715718	3
0.767876	3.333333333
0.815102	3.666666667
0.857504	4
0.895244	4.333333333
0.928534	4.666666667
0.957621	5
0.98278	5.333333333
1.004308	5.666666667
1.022511	6
1.037705	6.333333333
1.0502	6.666666667

the boundary layer [6]. Outside of this region, the flow is almost uniform.

### 6.3 Lid-Driven Cavity Flow

We investigate flow in a lid-driven cavity. We use the no-slip condition on the four boundaries with the lid having a constant velocity of  $(1, 0)$ . The computational domain has dimension  $1 \times 1$  subdivided into equally spaced subintervals in both the  $x$  and  $y$  directions. The Reynolds number used for this flow is 100.

When the parameters are chosen as above, these regions of recirculation known as eddies are formed in the two lower corners of the cavity.

### 6.4 Flow Over a Backward Facing Step

We investigate flow over a backward facing step. We take the no-slip condition on the north and south boundaries and the inflow/outflow condition on the east and west boundaries. The initial inflow velocity in the horizontal direction is set to 1 and the vertical component is set to 0. The dimension of the domain is  $30 \times 1.5$  in which the  $x$ -direction is subdivided into 100 equally spaced subintervals and

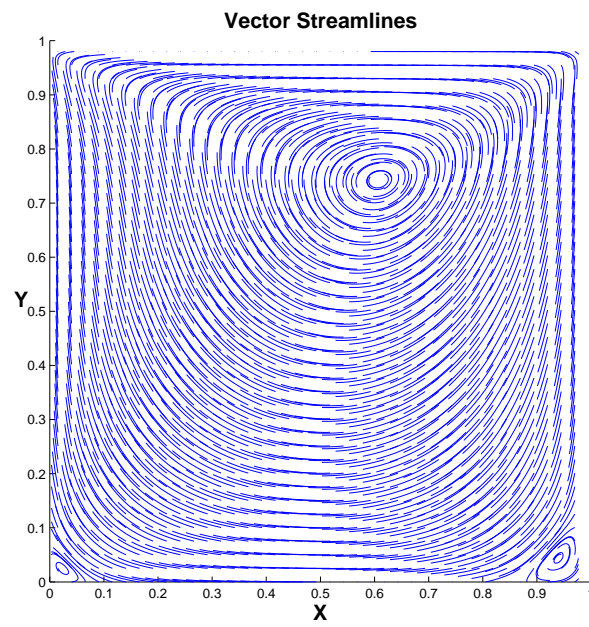


Figure 6.4: Flow in a Lid-Driven Cavity



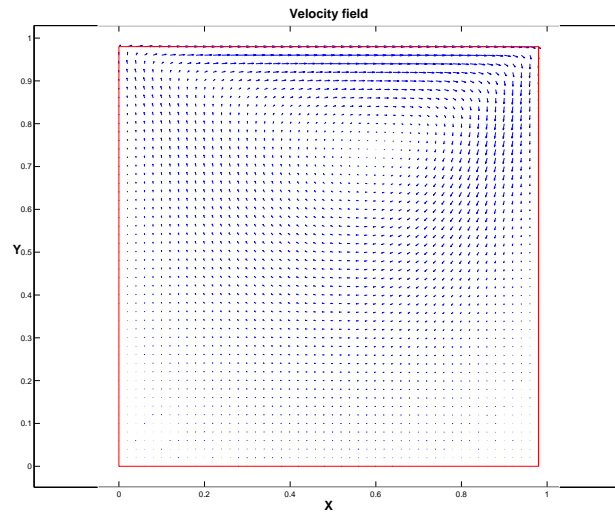


Figure 6.5: Flow in a Lid-Driven Cavity

the  $y$ -direction is subdivided into 25 equally spaced subintervals. The Reynolds number used for this flow is 500. It can be seen that there is a recirculation region behind the step as one expects.

## 6.5 Conclusion

In this chapter, flow between two parallel plates, flow in a lid-driven cavity, and flow over a backward facing step are investigated. The effects of different Reynold's numbers is also presented. We conclude by summarizing what has

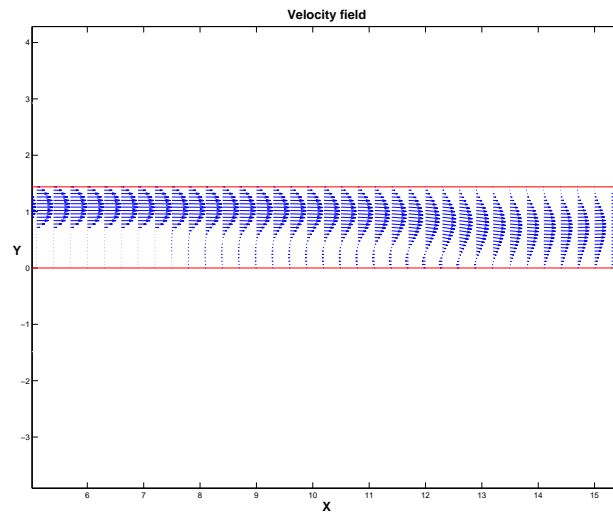


Figure 6.6: Flow Past a Backward Facing Step.

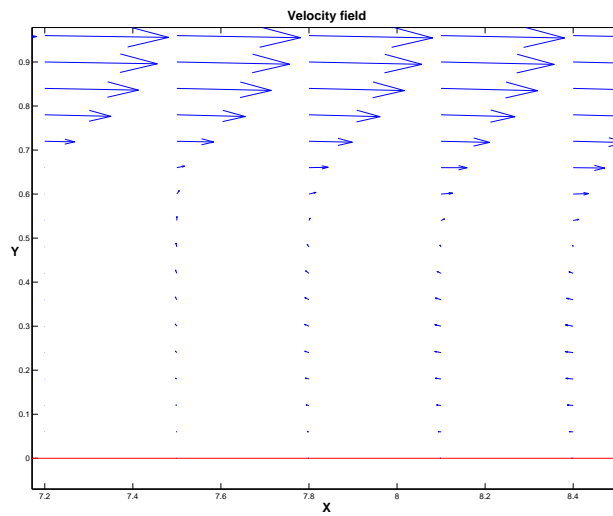


Figure 6.7: Flow Past a Backward Facing Step.

been done.

# Chapter 7

## Summary

In this thesis, we discussed numerical solutions of the 2-D Navier-Stokes equation. We began by laying the foundation needed for the presentation of this work. Reynold's Transport Theorem was proven and subsequently used in the derivation of the continuity equation and momentum equation from conservation of mass and linear momentum. From this point, dimensional analysis of the Navier-Stokes equations as well as the different types of boundary conditions needed in order to yeild a well-posed problem are investigated. After the completion of the derivation, we proceed to translating the continuous problem into its discrete form. In order to do this, we first introduce the concept of finite-difference approxima-

tions of continuous differential operators. The Navier-Stokes equations are then discretized and a technique for handling stability issues, known as the donor-cell scheme, is presented. We also mention that other popular methods for discretizing the Navier-Stokes equations are known. For example, Finite-Volume Method and Finite-Element Method for discretization. Chorin Projection Method is also discussed for the purpose of solving the Poisson Pressure Equation that arises after discretization. In chapter 3, the functions required for implementation of the algorithm for numerically solving Navier-Stokes equations are discussed. In the fourth chapter, visualization techniques are discussed. To this end, the concept of streamlines, streaklines, and velocity profiles are presented. Finally, we present examples of classical flow problems. The problems investigated are flow between two parallel plates, flow in a lid-driven cavity, and flow past a backward facing step. It is worth mentioning that when working with the two-dimensional Navier-Stokes equations, the Stream Function-Vorticity formulation can be used to reduce the number of parameters present by eliminating the presence of the pressure term. However, in the extension to three dimensions, this effect is lost. As was stated earlier, solutions to Navier-Stokes equations are well known in 2-D but there is much left to be done in 3-D. This thesis helps to lay the foundation so the more complicated yet interesting problems can be addressed.

# References

# Bibliography

- [1] Chorin A J 1968 Numerical Solution of the Navier-Stokes Equations, Mathematics of Computation, Vol. 22, No. 104,745-762.
- [2] Griebel M, dornseifer T Neunhoeffler, T 1997 Numerical Simulation in Fluid Dynamics: A Practical Approach (Philadelphia: SIAM)
- [3] Munson B, Young D, Okiishi T 2008 Fundamentals of Fluid Mechanics (New York:John Wiley and Sons,Inc), Fifth Edition
- [4] Pozrikidis C 1998 Numerical Computation in Science and Engineering (Oxford: Univ. Press)
- [5] Mason D P 1997 Finite Elasticity Lecture Notes, University of the Witwater-

srand, Johannesburg, South Africa

[6] Mason D P 1997 Fluid Mechanics Lecture Notes, University of the Witwatersrand, Johannesburg, South Africa