7-8-2022

Navier-Stokes equations in one and two dimensions

Jon Nerdal
Louisiana State University and Agricultural and Mechanical College

NAVIER-STOKES EQUATIONS IN ONE AND TWO DIMENSIONS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Mathematics

in

The Department of Mathematics

by
Jon Nerdal
B.A., University of Minnesota, 2020
M.S., Louisiana State University, 2022
August 2022

Acknowledgments

I would like to thank a lot of people for helping me along the way.

To my advisor, Gestur Olafsson, for getting this project started, for trusting in me to explore a subject that is quite different from his usual work, and for having excellent explanations and guidance when I needed it.

To Dr. Shipman for assisting with initiating this project, and to Dr. Tarfulea and Dr. Nguyen for agreeing to be on my committee.

To Coach Dennis Shaver and Coach Greg Watson for believing in my athletic abilities and for having me be a part of the track and field team at LSU which allowed me to pursue this degree.

To Steffen, Willy and Signe for always supporting me and encouraging me to keep improving my skills and knowledge within all areas of my life.

To Emma for supporting me and helping me to become a better version of myself, and for countless hours of listening to me talk about this project.

To my friends for the conversations, great times and memories.

To LSU as a whole for being the perfect fit for me on every level; academically, athletically and personally.

For all of this I am eternally grateful.

Table of Contents

Abstract

The Navier-Stokes equations are an important tool in understanding and describing fluid flow. We investigate different formulations of the incompressible Navier-Stokes equations in the one-dimensional case along an axis and in the two-dimensional case in a circular pipe without swirl. For the one-dimensional case we show that the velocity approximations are remarkably accurate and we suggest that understanding this simple axial behaviour is an important starting point for further exploration in higher dimensions. The complexity of the boundary is then increased with the two-dimensional case of fluid flow through the cross section of a circular pipe, where we investigate two separate formulations of the Navier-Stokes equations and observe their differences. The first two-dimensional formulation exhibits an auxiliary field which differs from the velocity by a gauge transformation. We are then able to eliminate the ambiguity related to the pressure boundary condition in the traditional formulation since the gauge freedom lets us assign specific and simple boundary conditions for both the auxiliary field and the gauge field. The latter two-dimensional formulation considers external forces acting on the fluid and resembles a more traditional approach to solving the Navier-Stokes equations. The two-dimensional results are then discussed and found to correspond with fluid mechanics theory given the initial conditions as well as the boundary conditions of the systems.

Chapter 1. Introduction

The Navier-Stokes equations are named after Claude-Louis Navier (1822) and George Gabriel Stokes (1850) and are mathematical equations used to describe conservation of mass and momentum for fluids, more specifically Newtonian fluids. To have a complete equation set we also need an equation of state relating pressure, temperature and density [WS12]. We obtain the equations by applying Newton's second law to the motion of a fluid, while assuming that the stress within the fluid is equivalent to the sum of a diffusing viscous term and a pressure term. They differ from Euler equations in the way that Euler equations only model inviscid flow, whereas the Navier-Stokes equations consider viscosity. Therefore, the Navier-Stokes equations possess better analytic properties and is preferred when computing numerical results for fluid flow [BCM01].

The Navier-Stokes equations are a set of partial differential equations used to describe the movement of viscous fluids, in general they describe how fluids move around an object [AD99] [LL59]. The equations provide accurate approximations of the velocity and the direction of the velocity within the fluid. When we are discussing the flow of a fluid there are a few things that needs to be considered. First of all, we need to determine whether the fluid is incompressible or compressible. If the molecules within the fluid are compressible it is very difficult to describe what happens to the fluid when variables such as velocity and pressure changes. This leads to a very complex system that we would prefer to avoid dealing with. Being incompressible on the other hand would entail that the density of the fluid is constant, and is a much simpler system to deal with. The Navier-Stokes equations used in this project are incompressible, as most of the calculations can be accurately done without considering changes in the fluids density. Another important

aspect of describing the fluid is the state of the fluid flow. For instance, there is a great difference between turbulence and laminar flow. We will focus on laminar flow, which is to say that the flow of the fluid is fully developed [PA17].

Besides being accurate approximations, the Navier-Stokes equations can be estimated fairly quickly. This means that we can achieve approximations that are very accurate while the runtime of the computations is low. In addition to this important benefit, we can apply the Navier-Stokes equations to a wide range of surfaces and objects [AC17] [BSDHO21] [RH21] [BGWX22]. The one-dimensional case discussed in this paper applies the equations to an axis [SO13], while the two-dimensional cases describe the flow of a fluid through a circular pipe. The reason for choosing the circular pipe is that it is very applicable to real life situations, including pumping and transporting oil [MMN21], blood flow within arteries [RH21] and transport of water through pipes [TB20]. Although there has been some research done using arbitrary shapes for the cross section of the pipe [BSDHO21] [RH21] not much has been done looking at the circular cross section specifically. We will assume the fluid has no swirl, which then makes the problem comparable to a two-dimensional fluid admitting an axial symmetry [VIL22] and suggests that a circular cross section will describe the fluid flow accurately. Most papers published on this topic revolve around square matrices and therefore also square-shaped cross sections [EL03] [BCM01] [TO20] [XHBC21] [LD22]. In the section about two-dimensional cases we are also using square matrices, however we introduce the unit circle as a boundary for the system. Then we end up with a circular cross section which allows us to mimic a real life situation. We will consider different formulations of the equations [LL59] [EL03] [LA12] and discuss how the various terms and variables effect the approximations obtained.

Chapter 2. One-dimensional Navier-Stokes

The following equation [LL59] assumes that the viscosity coefficients of the fluid can be considered as constant:

$$\rho(\frac{\partial v}{\partial t} + (v \cdot \nabla)v) = -\nabla p + \eta \Delta v + (\zeta + \frac{1}{3}\eta)\nabla \cdot \text{div} v \tag{2.1}$$

If the fluid is incompressible, which entails that the fluid's density is constant, we have div $v = 0$. Then, the Navier-Stokes equation rewrites as follows:

$$\frac{\partial v}{\partial t} + (v \cdot \nabla)v = -\frac{1}{\rho}\nabla p + \frac{\eta}{\rho}\Delta v \tag{2.2}$$

Where η is the dynamic coefficient of an incompressible fluid. The incompressible flow assumption holds when the fluid is not relativistic, so that the velocity of the fluid is much smaller than the speed of sound. Additionally, the equation assumes that the fluid is a continuum and therefore does not consider phenomena that are described at the molecular level. Both the fluid pressure p and the fluid velocity v are unknowns in this equation, as well as the density ρ of the fluid. In this case we are working with a one-dimensional version of the Navier-Stokes equation. Later, in chapter 3, we will discuss the two-dimensional case. Now, we must apply the continuity equation in order to solve the Navier-Stokes equation:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho v) = 0 \tag{2.3}$$

Now we are left with a system of two equations and two unknowns. While solving this system still requires multiple steps, it is much easier to deal with:

$$\frac{\partial u}{\partial t} + u \cdot \frac{\partial u}{\partial x} = f(t) + c \cdot \frac{\partial u}{\partial^2 x} \tag{2.4}$$

Where we let $f(t) = e^{-at-b}$ and $u = (x, t)$ is the velocity of the fluid. Next, we introduce intermediate values x^*, u^* and t^* for x, u and t, respectively:

$$x^* = x - \frac{1}{a} \cdot t \cdot e^{-b} - \frac{1}{a^2} \cdot e^{-at-b} + \frac{1}{a^2} 1/a^2 \cdot e^{-b} \tag{2.5}$$

$$u^* = u - (\frac{1}{a} \cdot e^{-b} - \frac{1}{a} \cdot e^{-at-b}) \tag{2.6}$$

$$t^* = t \tag{2.7}$$

The reason for introducing these intermediate variables is to simplify the expressions and to be able to rewrite u as follows:

$$u = u^* + \frac{1}{a} \cdot e^{-b} - \frac{1}{a} \cdot e^{-at-b} \tag{2.8}$$

At first sight this might not seem significant. However, writing the velocity u in this way allows us to rewrite equation 2.4, which we will show how to do in a few steps. The goal here is to arrive at an expression that can easily be implemented and calculated at each time step so that we can obtain an accurate solution to the equation. First we need to express the partial derivatives of the intermediate variables:

$$\frac{\partial u}{\partial x} = \frac{\partial u^*}{\partial x^*} \cdot \frac{\partial x^*}{\partial x} = \frac{\partial u^*}{\partial x^*} \tag{2.9}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u^*}{\partial x^{*2}} \tag{2.10}$$

$$\frac{\partial u}{\partial t} = \frac{\partial u^*}{\partial x^*} \cdot \frac{\partial x^*}{\partial t} + \frac{\partial u^*}{x^*} \cdot \frac{\partial x^*}{\partial x} + \frac{1}{a} \cdot e^{-at-b} \cdot (-a) \tag{2.11}$$

$$\frac{\partial x^*}{\partial t} = -\frac{1}{a} \cdot e^{-at-b} - \frac{1}{a^2} \cdot e^{-at-b} \cdot (-a) \tag{2.12}$$

$$\frac{\partial t^*}{\partial t} = 1 \tag{2.13}$$

Since we now know the different partial derivatives we can observe the relation we get between the partial derivative of u with respect to time:

$$\frac{\partial u}{\partial t} = \frac{\partial u^*}{\partial x^*} \cdot (-\frac{1}{a} \cdot e^{-b} + \frac{1}{a} \cdot e^{-at-b}) + \frac{\partial u^*}{\partial t^*} + e^{-at-b} \tag{2.14}$$

Then, we can again look at equation 2.4 and substitute in the relations we have derived for the intermediate variables:

$$\frac{\partial u^*}{\partial x^*} \cdot (-\frac{1}{a} \cdot e^{-b} + \frac{1}{a} \cdot e^{-at-b}) + \frac{\partial u^*}{\partial t^*} + e^{-at-b} + (u^* + \frac{1}{a} \cdot e^{-b} + \frac{1}{a} \cdot e^{-at-b}) \cdot \frac{\partial u^*}{\partial x^*} = e^{-a-b} + c \cdot \frac{\partial^2 u^*}{\partial x^{*2}} \tag{2.15}$$

Notice that we now can cancel some terms and simplify the equation so that we get this equation:

$$\frac{\partial u^*}{\partial t^*} + u^* \cdot \frac{\partial u^*}{\partial x^*} = c \cdot \frac{\partial^2 u^*}{\partial x^{*2}} \tag{2.16}$$

What remains now is to relate the partial derivatives to something we can calculate at each time step in a time iterating scheme:

$$u^*(t, x) \approx u^*(t_n, x_i) = u_i^{*n} \tag{2.17}$$

$$\frac{\partial u^*}{\partial x} = u_x^* \approx \frac{(u_{i+1}^* - u_{i-1}^*)}{2\Delta x} \tag{2.18}$$

$$\frac{\partial u^*}{\partial^2 x} = u_{xx}^* \approx \frac{(u_{i+1}^{*n} - 2u_i^{*n} + u_{i-1}^{*n})}{(\Delta x)^2} \tag{2.19}$$

$$\frac{\partial u^*}{\partial t} = u_t^* \approx \frac{(u_i^{*n+1} - u_i^{*n})}{\Delta t} \tag{2.20}$$

Then, equation 2.4 rewrites as:

$$\frac{u_i^{*n+1} - u_i^{*n}}{\Delta t} + u_i^{*n} \cdot \frac{u_{i+1}^* - u_{i-1}^*}{2\Delta x} = c \cdot \frac{u_{i+1}^{*n} - 2u_i^{*n} + u_{i-1}^{*n})}{(\Delta x)^2} \tag{2.21}$$

Which leaves us with a system that is fairly simple to calculate using computer programming:

$$u_i^{*n+1} = u_i^{*n} - u_i^{*n} \cdot \Delta t \cdot \frac{u_{i+1}^{*n} - 2u_i^{*n} + u_{i-1}^{*n}}{(\Delta x)^2} + c \cdot \Delta t \cdot \frac{u_{i+1}^{*n} - 2u_i^{*n} + u_{i-1}^{*n}}{(\Delta x)^2} \tag{2.22}$$

The equation above is then converted to MatLab code and iterated over time so we can get a three dimensional plot of the approximated solution to the Navier-Stokes equations. We will also approximate a solution with boundary conditions $u = 0$ to investigate how well suited the Euler equation is as a boundary condition for velocity. The reason for doing this is that the most exact solution we can get of the system is achieved by setting $u = 0$ as a boundary condition. Although it is still an approximation of the velocities, this formulation will serve as an optimal solution and can then be utilized for calculating error. Figures 2.1 and 2.2 exhibit the results for the one-dimensional Navier-Stokes equations:

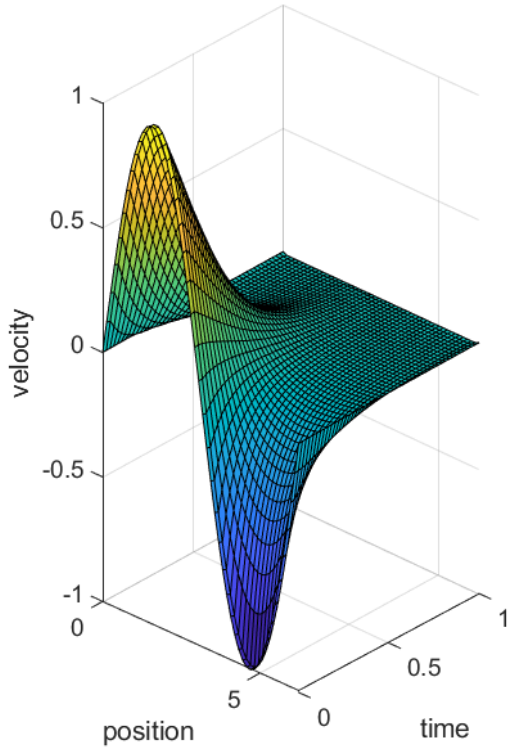

Figure 2.1. Solution to the one-dimensional Navier-Stokes equation with initial function $f = \sin(x)$, a = 2, b = and c = 0.1, Euler equation $\frac{1}{a} \cdot e^{-b} - \frac{1}{a} \cdot e^{-a \cdot t - b}$ is used on the boundaries $x = 0$ and $x = 2\pi$. The x-, y- and z-axes are position, time and velocity, respectively. The matrix used is 50x50, so we have 50 points along the position and time axes.

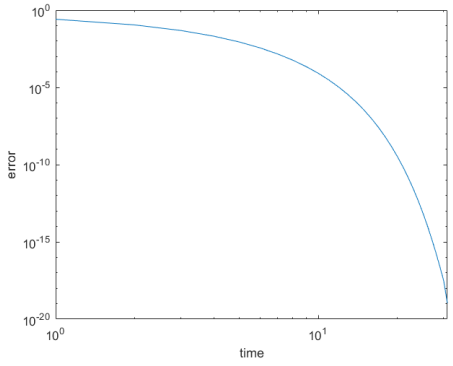

Figure 2.2. Logarithmic plot of the error of the solution presented in figure 2.1. Figure 2.1 is compared with a solution that has boundary condition $u = 0$ at $x = 0$ and $x = 2\pi$ instead of using Euler equation $\frac{1}{a} \cdot e^{-b} - \frac{1}{a} \cdot e^{-a \cdot t - b}$ as boundary condition.

From figure 2.1 we can see that initially the sine function $f = sin(x)$ determines the velocity of the fluid and the velocity is distributed exactly as the sine function we are applying. Since this is the one-dimensional case, all iterations over time only considers a line along the position axis. Imagine the position axis being the one dimensional boundary for the fluid at time $t = 0$. Then as time increases we move along the time axis while still only considering a single line in the position direction. This line represents the velocities of the system at the given time t. When we remove the initial force, we see that the fluid stabilizes for each iteration over time and approaches 0 everywhere along the position axis. When the velocities reach 0 everywhere the impact of the initial force is completely lost. Figure 2.2 shows a logarithmic plot of time vs error. We got this plot from comparing the values in figure 2.2 with a solution that has $u = 0$ as the boundary conditions instead of $u = e^{-at-b}$ which is Euler's formula. Since the graph curves downwards we can obtain that the error converges to 0 faster than logarithmic time from around time $t = 10$. This is a significant result and is an indication that the approximation method for the one dimensional Navier-Stokes equations is very good as we need few iterations over time to get an accurate approximation of the system.

Chapter 3. The two-dimensional case

In this section we will investigate the Navier-Stokes equation in two dimensions, so here we will use $u = (u, v)$ for the velocity of the fluid. For the calculations we will consider a circular cross section of a pipe to simulate flow through a pipe which is applicable to a wide range of problems within fluid dynamics. Typical situations where circular cross sections are important include pumping and transportation of oil [MMN21], water flow through pipes [TB20] and blood flow through arteries [RH21]. The domain of the cross section will be the unit circle.

3.1. Gauge Formulation

The gauge formulation approach of the incompressible Navier-Stokes equation is demonstrated to be well suited for numerical calculations [EL03]. We start with the classical version of the Navier-Stokes equation given as follows:

$$u_t + (u \cdot \nabla)u + \nabla p = \frac{1}{Re} \cdot \Delta u \tag{3.1}$$

$$\nabla \cdot u = 0 \tag{3.2}$$

Applied to the fluid Ω, where $u = (u, v)$ is the velocity of the fluid and p is the pressure. The standard simple boundary condition is:

$$u = 0 \tag{3.3}$$

Which is applied to the boundary of the fluid, $\partial\Omega$. In the case of flow through a pipe, the boundary of the fluid would be in contact with the inner walls of the pipe since we are assuming that the internal flow in the pipe is fully developed [BW78].

Now, by taking the divergence of the momentum equation, we get a Poisson equation for the pressure within the fluid:

$$-\Delta p = \nabla(u \cdot \nabla u) \tag{3.4}$$

Applying this Poisson equation directly to the Navier-Stokes equation we can substitute out the incompressibility condition [EL03]. However, getting rid of the incompressibility condition also means that we are removing the boundary condition presented earlier in equation 3.3. Therefore, we need to derive a new boundary condition that agrees with the Poisson equation. One option would be to extend the Navier-Stokes equation to the boundary:

$$\frac{\partial p}{\partial n} = n \cdot \Delta u \tag{3.5}$$

The equation above requires that we evaluate the viscous term at the boundary, which is very difficult to do precisely. Keeping consistency between the Poisson equation 3.4 and equation 3.5 is also an issue since it is a Neumann problem in a discrete setting [EL03]. To work around the difficulty related to the pressure boundary condition a projection method was invented. The disadvantage with this projection method is that we must use special discretization schemes to discretize the pressure Poisson equation, which limits both the simplicity and the flexibility of the projection method [EL03].

For the gauge method we first want to replace pressure by a gauge variable ϕ and then introduce an auxiliary field $a = u + \nabla\phi$, and then replace the classical Navier-Stokes equation by

$$a_t + (u \cdot \nabla)u = \frac{1}{Re} \cdot \Delta a \tag{3.6}$$

$$\Delta\phi = \nabla \cdot a \tag{3.7}$$

The equations above are clearly equivalent to the initial version of the Navier-Stokes equation, and we can see that ϕ is related to the pressure by:

$$p = \partial_t \phi - \frac{1}{Re}\Delta\phi \tag{3.8}$$

The greatest advantage of using this formulation is that we can use the gauge freedom to assign an unambiguous boundary condition for a and ϕ. We prescribe the following:

$$\frac{\partial\phi}{\partial n} = 0, a \cdot n = 0, a \cdot \tau = \frac{\partial\phi}{\partial\tau} \tag{3.9}$$

Or:

$$\phi = 0, a \cdot n = \frac{\partial\phi}{\partial n}, a \cdot \tau = 0 \tag{3.10}$$

Applied to the boundary. τ is the unit vector in the tangential direction, and n is the unit vector in the normal direction.

The equations presented in 3.6-3.10 can be solved by various numerical schemes. We will focus on a few different approaches that include the Neumann boundary conditions described for the gauge field [EL03]. Where this paper differs from previous findings is that we will focus on a circular cross section. Since we are using numerical methods

that calculates the velocity at each time step, we will also need to solve the Poisson equation at each time step.

The first approach for solving the Navier-Stokes equation numerically is by using the backward Euler in time scheme. In that case the finite difference scheme can be written as [EL03]:

$$\frac{a^{n+1} - a^n}{\Delta t} + (u^n \cdot \nabla_h) \cdot u^n = \frac{1}{Re} \cdot \Delta_h \cdot a^{n+1} \tag{3.11}$$

$$\Delta_h \cdot \phi^{n+1} = \nabla_h \cdot a^{n+1} \tag{3.12}$$

$$u^{n+1} = a^{n+1} - \nabla_h \cdot \phi \tag{3.13}$$

Rearranging the expression, we get:

$$a^n = \Delta t \cdot a^{n+1} + \Delta t \cdot (u^n \cdot \nabla_h) u^n - \frac{\Delta t}{Re} \cdot \Delta_h \cdot a^{n+1} \tag{3.14}$$

We choose to iterate increasingly over time starting at $t = 0$, so we will use the forward Euler in time scheme which is derived similarly to backward Euler:

$$\frac{a^{n+1} - a^n}{\Delta t} + (u^n \cdot \nabla_h) \cdot u^n = \frac{1}{Re} \cdot \Delta_h \cdot a^n \tag{3.15}$$

$$\Delta_h \cdot \phi^{n+1} = \nabla_h \cdot a^{n+1} \tag{3.16}$$

$$u^{n+1} = a^{n+1} - \nabla_h \cdot \phi \tag{3.17}$$

So that the two latter equations remain unchanged. Like we did with the backward Euler in time scheme, we can rearrange the expression as follows:

$$a^{n+1} = \Delta t \cdot a^n + \Delta t \cdot (u^n \cdot \nabla_h) u^n - \frac{\Delta t}{Re} \cdot \Delta_h \cdot a^n \tag{3.18}$$

Now we have a scheme where we first distribute the initial velocity u_0 over the cross section of the pipe. Using a matrix that covers the unit square, we let the initial velocity u_0 be of the form:

$$u_0(x,y) = \begin{cases} cos(\frac{\pi}{2} \cdot ||x^2 + y^2||), & \text{if } x^2 + y^2 \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.19}$$

The equation above initializes the velocity in the pipe for a viscous fluid [BW59], and if we initialize the pressure for our system via the gauge variable as $\phi_0 = 0$, we get that at $a_0 = u_0$ where a_0 is the initialization of the auxiliary field a. We can then create a separate system with a higher initial pressure, so that the variable $\phi \neq 0$. This will allow us to determine the effects the initial pressure impose on the velocities in the pipe. For the second system, we will keep the same initial velocity as presented in equation 3.19 combined with the following initialization for the gauge variable ϕ:

$$\phi_0(x,y) = \begin{cases} -cos(\frac{\pi}{2} \cdot ||x^2 + y^2||), & \text{if } x^2 + y^2 \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.20}$$

After this first step we will calculate $a^{n+1}(x,y)$ from equation 3.18 at each time step, and then update the gauge variable $\phi(x,y)$ from equation 3.16. At the end of each iteration we will update the velocities $u^{n+1}(x,y)$ for each point (x, y) in the matrix. Since

the domain is within the pipe we will only focus on the points that are within the unit circle. The points that are outside of the unit circle will be set to be 0 for all variables.

One challenge that arises from setting the boundary to be the unit circle is that we need to calculate the gradient ∇_h and the Laplacian Δ_h for points on the boundary when there are many points within the matrix that are outside of the domain of the pipe. ∇_h and Δ_h are calculated by using the data at surrounding points, which means that for boundary points we could end up using data from some points of the matrix that are outside of the unit circle and therefore outside of our domain. This would be avoided if we wanted to look at the unit square, but since we want to look at a circular cross section we need to make some adjustments. A way to deal with this issue is to introduce intermediate variables for the pressure and for the velocity, and set the points that are outside of the domain that we will use in the calculations to have the same value as the point on the boundary that is closest to that particular point. That way we get an accurate approximation while discarding the intermediate values that lie outside the domain of the pipe. The way we solve this is quite simple and is presented in the appendix B. These are the contour plots we get for the cross section of the pipe using the gauge formulation, where the only differences between the two systems is the initial gauge variable ϕ:

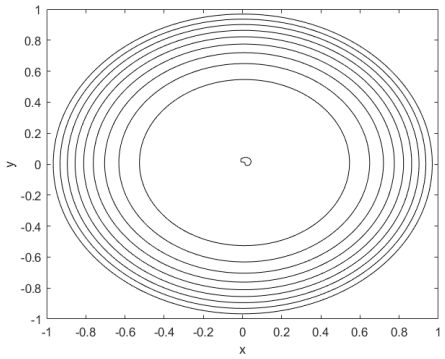

Figure 3.1. Contour plot of the solution of the two-dimensional Navier-Stokes equations using the gauge formulation at $t = 10$ with $\mathrm{Re} = 40000$. Initial conditions are u_0 as presented in equation 3.19 and initial pressure is zero everywhere. Each contour line represents a difference in velocity of 0.1.

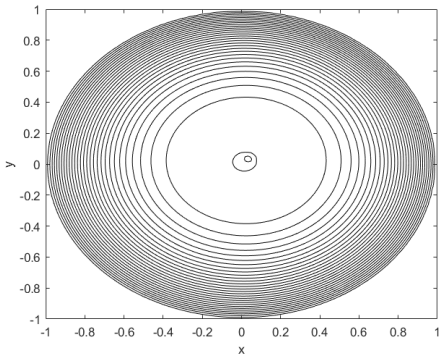

Figure 3.2. Contour plot of the solution of the two-dimensional Navier-Stokes equations using the gauge formulation at $t = 10$ with $\mathrm{Re} = 40000$. Initial conditions are u_0 as presented in equation 3.19 and initial value of ϕ is as described in equation 3.20. Each contour line represents a difference in velocity of 0.1.

Observing the contour plot we see that the flow through the pipe is smooth and continuous since there are no irregularities in the contour lines. The issue concerning irregularities often occur on the boundary, and we can see that the boundary is smooth for both figures 3.1 and 3.2. Towards the center of both contour plots it is clear that not a perfect circle is achieved for the innermost contour line. These differences in velocity towards the center could be due to the direction in which we update the matrix, as we update it from left to right. Another possibility that could correlate with the slight deviation for the innermost contour line is the approach for calculating the gradient at the boundary of the unit circle. This calculation would result in slight changes in the surrounding points, and as we iterate through the matrix the velocities towards the center would be slightly altered as well.

From figures 3.1 and 3.2 it is apparent that a lower initial value for the gauge variable ϕ results in a higher velocity of the system. Increasing ϕ would directly increase the pressure p as well, and their relation is described in equation 3.8. The initial value for ϕ_0 is set to be negative for the second system as described in equation 3.20, and we can observe the differences this gives for the contour plot; the system without initial pressure has about 10 contour lines while the second system with a negative initial pressure has about 30 contour lines.

Next, we look at the velocity matrices obtained using the gauge method that correspond to the contour plots presented in figures 3.1 and 3.2:

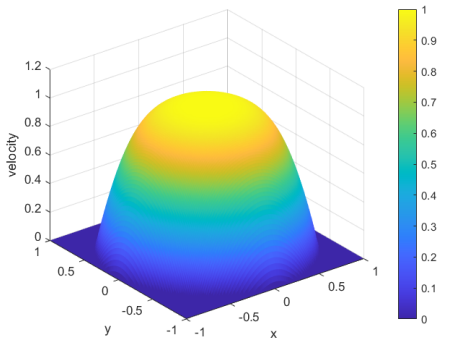

Figure 3.3. Solution of the two-dimensional Navier-Stokes equations using the gauge formulation at $t = 10$ with Re $= 40000$. Initial conditions are u_0 as presented in equation 3.19 and initial pressure is zero everywhere. The matrix used for this plot is 150x150. The contour plot of this solution is given in figure 3.1.

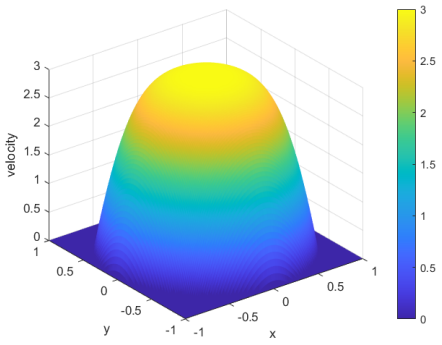

Figure 3.4. Solution of the two-dimensional Navier-Stokes equations using the gauge formulation at $t = 10$ with Re $= 40000$. Initial conditions are u_0 as presented in equation 3.19 and initial value of ϕ is as described in equation 3.20. The matrix used for this plot is 150x150. The contour plot of this solution is given in figure 3.2.

The figures 3.3 and 3.4 show more clearly the velocities within the pipe for the two systems we have approximated using the gauge transformation. The part of the matrices that lie outside of the unit circle has the velocity set to 0 and therefore we get the flat parts in the corners since they are not within the pipe. When looking at the figures these corners should not be interpreted as they are not within the domain of our system and are purely there for computational reasons as shown in appendix B.

The figures 3.1 - 3.4 tell us that there are no singularities in the matrices, and also that the fluid flow for the system without an initial pressure has velocities that are very similar to the initial velocity u_0 throughout the cross section. Decreasing the initial pressure turns out to increase the final velocity by approximately a factor of 3.

3.2. Two-dimensional Navier-Stokes with forces

The Navier-Stokes equation used in the previous section does not take body forces such as gravity into account. To study the impact that forces have on a system of fully developed flow, laminar flow, we will use this section to investigate a different version of the Navier-Stokes equation where forces are included [LA12]:

$$u_t + (u \cdot \nabla)u = -\frac{1}{\varrho} \cdot \nabla p + \nu \cdot \nabla^2 u + f \tag{3.21}$$

The equation above will be used along with the mass balance equation:

$$\nabla \cdot u = 0 \tag{3.22}$$

Where ϱ is the density of the fluid, ν is the kinematic viscosity and f is forces such as gravity. This version of the Navier-Stokes equation looks a little different than the version discussed in the previous section, so we provide some background for how we arrive at this particular formulation. Using the general momentum balance equation for a continuum, from Newton's second law of motion, we obtain [LA12]:

$$\varrho \frac{Du}{dt} = \nabla \cdot \omega + \varrho f \tag{3.23}$$

Where ω is the stress tensor and the operator $\frac{D}{dt}$ is the material derivative which can denote acceleration as follows:

$$\frac{Du}{dt} = u_t + (u \cdot \nabla)u \tag{3.24}$$

In other words, $\varrho\frac{Du}{dt}$ is density times acceleration, while $\nabla \cdot \omega + \varrho f$ are the internal stresses and external forces that induce the motion u [LA12].

Next, we look at another fundamental equation for fluids, namely the continuity equation. The general form of the continuity equation is given as:

$$\varrho_t + \nabla \cdot (\varrho u) = 0 \tag{3.25}$$

Which we can rewrite to get the following relation [LA12]:

$$\nabla \cdot u = \frac{1}{\varrho}\frac{D\varrho}{dt} \tag{3.26}$$

Since the fluid is incompressible, the density of the fluid is constant. Applying this to equation 3.26 we get that $\frac{D\varrho}{dt} = 0$ since this is the rate of change of the density, and this leads to $\nabla \cdot u = 0$. Different types of fluids will have different relations between the motion u and the internal stresses ω. For instance, a Newtonian fluid has an isotropic, linear relation between u and ω [LA12]:

$$\omega = -pI + \mu(\nabla u + (\nabla u)^T) \tag{3.27}$$

where I is the identity tensor and μ is the dynamic viscosity of the fluid. Now, we can insert equation 3.27 into equation 3.23 to arrive at the Navier-Stokes equation 3.21. In order to do this we need to divide the expression by ϱ, setting $\nabla\cdot(Ip) = \nabla p$ and expressing $\nabla \cdot (\nabla u + (\nabla u)^T)$ as $\nabla^2 u + \nabla(\nabla \cdot u) = \nabla^2 u$.

Similar to what we did for the 2-dimensional case using the gauge method, we will approach this system using the Forward Euler scheme. However, we will make a few

changes to the scheme as we want to evaluate the pressure at each time step. Initially we start with the following equation:

$$u^{n+1} = u^n - \Delta t(u^n \cdot \nabla)u^n - \frac{\Delta t}{\varrho}\nabla p^{n+1} + \Delta t\nu\Delta^2 u^n + \Delta t f^n \tag{3.28}$$

Along with:

$$\nabla \cdot u^{n+1} = 0 \tag{3.29}$$

In order to solve this system at each time step we need to introduce an intermediate variable, call that variable u^* and let it be an intermediate variable for the velocity at a given point (x, y) in the cross section of the pipe. As we did before we will look at the unit square matrix and then let the pipe be the unit circle. First, we present an equation for the intermediate velocity variable u^*:

$$u^* = u^n - \Delta t(u^n \cdot \nabla)u^n - \beta\frac{\Delta t}{\varrho}p^n + \Delta t\mu\nabla^2 u^n + \Delta t f^n \tag{3.30}$$

We need to make a small adjustment to this equation (above) since u^* violates the incompressibility condition. Therefore we introduce a correction δu as follows:

$$u^{n+1} = u^* + \delta u \tag{3.31}$$

Such that $\nabla \cdot u^{n+1} = 0$. Next, we can reconstruct a few of the previous equations to obtain an expression for δu:

$$\delta u = u^{n+1} - u^* = -\frac{\Delta t}{\varrho}\nabla\phi \tag{3.32}$$

Where ϕ is introduced as a pressure change at each time step and we define ϕ as the following relation:

$$\phi = p^{n+1} - \beta p^n \tag{3.33}$$

Note that we will use $\beta = 0$, so the pressure change ϕ is not needed for our calculations. Now we explore what this entails for the incompressibility condition of the fluid, replacing u^{n+1} with $u^* + \delta u$:

$$\nabla \cdot (u^* + \delta u) = 0 \tag{3.34}$$

Then we use equation 3.32 to rearrange the expression. Doing this leaves us with a Poisson equation of the form:

$$\nabla^2 \phi = \frac{\varrho}{\Delta t} \nabla \cdot u^* \tag{3.35}$$

We will use Fast Fourier Transform (FFT) to compute ϕ in the Poisson equation above. The background information needed for FFT is included in the appendix A, and this is the material we are utilizing. In practice, we will use the built-in MATLAB functions for FFT and inverse FFT since we have evidence that these methods are very accurate [SM19]. The built-in functions are fast and accurate due to MATLAB's ability to choose the FFT verion most suitable for the data at hand. Obtaining the information about ϕ by using the FFT allows us to update the velocity at each time step, as well as updating the pressure at each time step. For the velocity update we get:

$$u^{n+1} = u^* - \frac{\Delta t}{\varrho}\nabla\phi \tag{3.36}$$

And the calculation we do to update the pressure is done by solving this equation:

$$p^{n+1} = \phi + \beta p^n \tag{3.37}$$

Now we have a scheme where we first distribute the initial velocity u_0 over the cross section of the pipe. Using a matrix that covers the unit square, we let the initial velocity u_0 be of the form:

$$u_0(x,y) = \begin{cases} cos(\frac{\pi}{2} \cdot ||x^2 + y^2||), & \text{if } x^2 + y^2 \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.38}$$

Equation 3.38 initializes the velocity in the pipe for a viscous fluid [BW59], which is the same initial velocity as we used in the previous section with the gauge formulation. In our calculations we will set $\beta = 0$, which omits the pressure term completely from equation 3.30, although there is no reason why the formulation will work just as well for any other β. To fully determine the influence of external forces on the system we will compare two systems with the same initial velocity u_0 from equation 3.38 but with different forces acting on the systems. We will also see how differences in initial pressure changes the velocities in the pipe. Therefore we will have four separate systems. For the first and the third system we will have the external forces f defined as follows:

$$f(x,y) = \begin{cases} 0.1 \cdot \cos(\frac{\pi}{2} \cdot ||x^2 + y^2||), & \text{if } x^2 + y^2 \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.39}$$

While for the second and the fourth system we will set the external forces f to be the following:

$$f(x,y) = \begin{cases} \cos(\frac{\pi}{2} \cdot ||x^2 + y^2||), & \text{if } x^2 + y^2 \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.40}$$

Now, for the first and the second system we will have the initial pressure set to be 0 everywhere, while for the third and the fourth system we will distribute the initial pressure as follows:

$$p(x,y) = \begin{cases} -\cos(\frac{\pi}{2} \cdot ||x^2 + y^2||), & \text{if } x^2 + y^2 \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.41}$$

Similarly to the velocity u_0, the points that are outside of the unit circle will have external forces $f = 0$ and pressure $p = 0$. One challenge that remains in regard to the velocity u_0 is that we need to calculate the gradient ∇_h and the Laplacian Δ_h for points on the boundary. ∇_h and Δ_h are calculated by using the data at surrounding points, which means we end up using data from points outside of the domain since there are points within the matrix that are not within the unit circle. This would be avoided if we wanted to look at the unit square, but since we want to look at a circular cross section we need to make some adjustments. A way to deal with this issue is to introduce intermediate variables for the pressure and for the velocity, and set the surrounding points that we will

use in the calculations to have the same value as the point on the boundary that is closest to that particular point. That way we get an accurate approximation while discarding the intermediate values that lie outside the domain of the pipe. This is the same procedure as described in chapter 3.1 and is displayed in further detail in appendix B.

These are the contour plots we get for the two-dimensional formulation with forces for the four different systems:

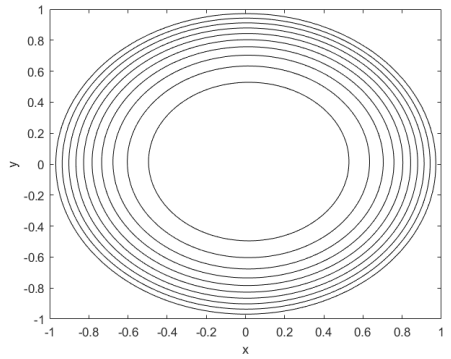

Figure 3.5. Contour plot of system 1 using the two-dimensional Navier-Stokes equations using the formulation with forces at $t = 10$ with kinematic viscosity $\nu = 1$ and Re $= 40000$. Initial conditions are u_0 as presented in equation 3.38, initial pressure is 0 and the force acting on the system is described in equation 3.39. Each contour line represents a difference in velocity of 0.1.

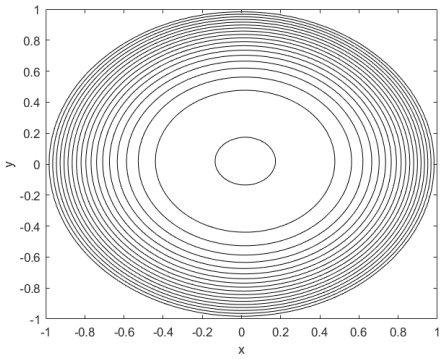

Figure 3.6. Contour plot of system 2 using the two-dimensional Navier-Stokes equations using the formulation with forces at $t = 10$ with kinematic viscosity $\nu = 1$ and Re $= 40000$. Initial conditions are u_0 as presented in equation 3.38, initial pressure is 0 and the force acting on the system is described in equation 3.40. Each contour line represents a difference in velocity of 0.1.

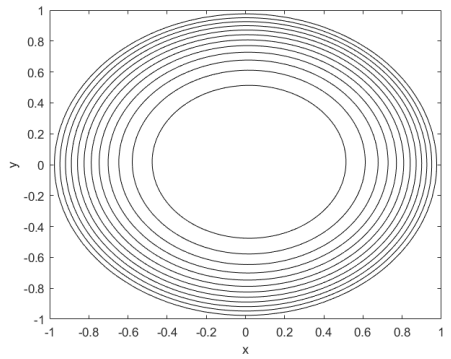

Figure 3.7. Contour plot of system 3 using the two-dimensional Navier-Stokes equations using the formulation with forces at $t = 10$ with kinematic viscosity $\nu = 1$ and Re $= 40000$. Initial conditions are u_0 as presented in equation 3.38, initial pressure as presented in equation 3.41 and the force acting on the system is described in equation 3.39. Each contour line represents a difference in velocity of 0.1.

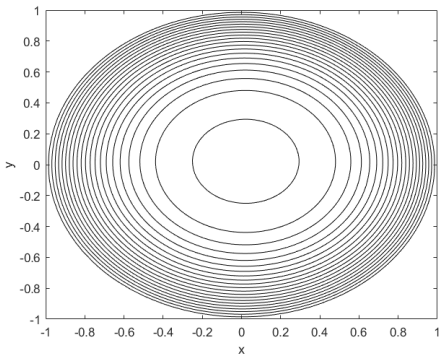

Figure 3.8. Contour plot of system 4 using the two-dimensional Navier-Stokes equations using the formulation with forces at $t = 10$ with kinematic viscosity $\nu = 1$ and Re $= 40000$. Initial conditions are u_0 as presented in equation 3.38, initial pressure as presented in equation 3.41 and the force acting on the system is described in equation 3.40. Each contour line represents a difference in velocity of 0.1.

Observing the contour plots we see that the flow through the pipe is smooth and continuous since there are no irregularities in the contour lines. The issue concerning irregularities often occur on the boundary, and we can see that the boundary is smooth for all figures 3.5 - 3.8. Towards the center of all the contour plots we see that the contour lines maintain a nearly perfect circle. In other words, the final velocities seem to be distributed similarly to the initial velocities throughout the cross section.

From figures 3.5 - 3.8 it is apparent that a lower force acting on the system results in lower velocities and that a lower initial pressure results in higher velocities. Therefore, the system with the highest velocity at time $t = 10$ is the system with a higher force acting on it and a lower initial pressure. We can also see that the force contributes more to the change of velocity than what the initial pressure does. Further inspection of the contour plots of the systems show that they have a different number of contour lines; system 1 has 10, system 2 has 18 , system 3 has 12 and system 4 has 20.

Next, we look at the velocity matrices obtained using the formulation including forces that correspond to the contour plots presented in figures 3.5 - 3.8:

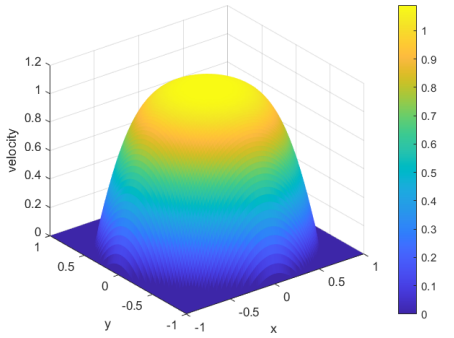

Figure 3.9. Solution of system 1 using two-dimensional Navier-Stokes equations using the formulation with forces at $t = 10$ with kinematic viscosity $\nu = 1$ and Re $= 40000$. Initial conditions are u_0 as presented in equation 3.38 and the force acting on the system is described in equation 3.39. The matrix used for this plot is 100x100. The contour plot of this solution is given in figure 3.5.

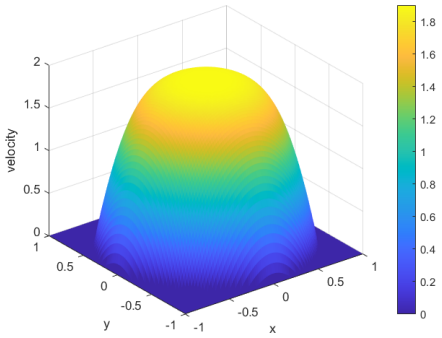

Figure 3.10. Solution of system 2 using two-dimensional Navier-Stokes equations using the formulation with forces at $t = 10$ with kinematic viscosity $\nu = 1$ and Re $= 40000$. Initial conditions are u_0 as presented in equation 3.38 and the force acting on the system is described in equation 3.40. The matrix used for this plot is 100x100. The contour plot of this solution is given in figure 3.6.

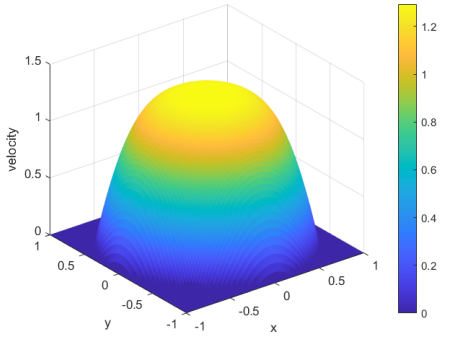

Figure 3.11. Solution of system 3 using two-dimensional Navier-Stokes equations using the formulation with forces at $t = 10$ with kinematic viscosity $\nu = 1$ and Re $= 40000$. Initial conditions are u_0 as presented in equation 3.38, initial pressure as presented in equation 3.41 and the force acting on the system is described in equation 3.39. The matrix used for this plot is 100x100. The contour plot of this solution is given in figure 3.7.

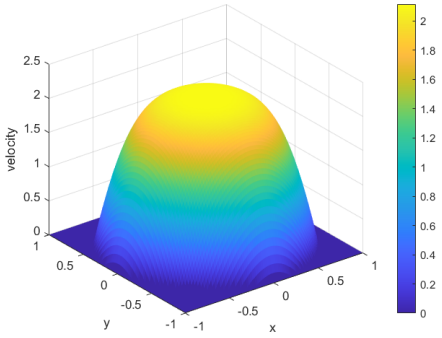

Figure 3.12. Solution of system 4 using two-dimensional Navier-Stokes equations using the formulation with forces at $t = 10$ with kinematic viscosity $\nu = 1$ and Re $= 40000$. Initial conditions are u_0 as presented in equation 3.38, initial pressure as presented in equation 3.41 and the force acting on the system is described in equation 3.40. The matrix used for this plot is 100x100. The contour plot of this solution is given in figure 3.8.

The figures 3.9 - 3.12 show more clearly the velocities within the pipe for the two systems we have approximated using the Navier-Stokes equations with external forces. The part of the matrices that lie outside of the unit circle has the velocity set to 0 and therefore we get the flat parts in the corners since they are not within the pipe. When looking at the figures these corners should not be interpreted as they are not within the domain of our system and are purely there for computational reasons as shown in appendix B.

We can obtain that all systems 1-4 have velocities close to 0 at the boundary, whereas system 4 has a higher velocity than the other systems due to a higher force acting on it and a lower initial pressure. The figures 3.5 - 3.12 tell us that there are no singularities in the matrices, and also that the fluid flow for the system with a lower force acting on the system has velocities that are very similar to the initial velocity u_0 throughout the cross section. Increasing the external force turns out to increase the final velocity by approximately a factor of 1.8, and decreasing the initial pressure changes the final velocity by a factor of about 1.15.

Chapter 4. Discussion

Examining the results from the different numerical methods we see that the pressure within the fluid and the external forces play a crucial role in how the fluid behaves. From the one-dimensional case we see how the velocities are distributed along the axis and how quickly the velocities converge towards 0 when the external force is removed. Furthermore, in the two-dimensional formulation using the gauge transformation, we observe that a lower pressure within the fluid results in a higher velocity. For smoothness of the system we prefer to distribute the pressure smoothly throughout the cross section of the pipe, which essentially means that the pressure of any given point p_0 is fairly similar to the pressure of its surrounding points. The same holds true for the distribution of external forces on the system. It is also apparent that applying a force to the system results in a greater change of velocity than changing the initial pressure. For the two-dimensional formulation considering forces we are setting $\beta = 0$ which effectively removes a pressure term in the equation. Using other constants for β could result in the final velocities being more dependent on the pressure within the fluid than what we have presented in chapter 3.2. Additionally, we set the dimensionless number $Re = 40000$ so that the calculations can be done regardless of unit preference [ST86].

There are a few challenges when it comes to creating an accurate model of the fluid flow through a pipe. The methods discussed in sections 3.1 and 3.2 use matrices to describe the fluid flow through a pipe, and for computational reasons the matrices are quite small compared to the number of molecules we would expect the fluid to contain. Therefore, we assign one point in the matrix grid to describe a large number of particle points in the fluid. In the figures presented in chapter 3 we are able to avoid the issues that would

arise since the radius of the pipe is fairly large and the variables such as velocity, pressure and forces are relatively low. However, if we wanted to investigate the effects of shortening the radius of the pipe we would get approximations that are more inaccurate than desired since each point in the matrix would then represent a very large portion of the fluid flow. Despite this flaw, the numerical solutions presented in this project are consistent with previous results [EL03] [LD22] which indicates that our calculations are accurate.

Exploring other research done on the topic we expect the presented formulations to be well suited for numerical approximations for a wide variety of complex, as well as non-complex, surfaces and initial conditions. For future research we suggest that the Navier-Stokes equations presented in this paper can be an important tool in calculating the velocities for fluid flow in a pipe depending on variables such as initial velocity, initial pressure and external forces. An area where this becomes of upmost importance is transportation of oil, where pumping and transporting requires an external force such as a pump. We can then calculate how much external force is needed to reach a certain fluid velocity, and also how much of the applied force directly results in higher velocities. That way we can optimize the cost of the force applied by the pumps compared to the velocity of the oil within the pipe. Using lower external forces increases the velocity of the fluid slightly while conserving more kinetic energy than a higher external force, even though the higher force results in higher velocities. That is to say there is a greater loss of energy when applying a high force to the system compared to using a lower force. Also, a very high external force runs the risk of creating turbulence if the velocities near the inner walls of the pipe are very high. This is because the boundary values for the velocity are almost 0, so there could be a great difference in velocities between two neighboring points close

to the boundary. There are other factors that play into this as well such as heat transfer [MMN21] and different viscosities within the fluid [BSDHO21]. The case of oil flowing through a pipe is a similar system to blood flowing through arteries; in both cases we have increased complexity compared to what is presented in our calculations. However, utilizing the Navier-Stokes equations from this paper is a great tool for approximations of real life systems.

Chapter 5. Conclusion

We have presented different versions of the Navier-Stokes equations in one and two dimensions and described how the axial and circular cases are relevant for real-life applications. The one-dimensional case yields an approximation that converges faster than logarithmic which is a desirable approximation and an indicator that starting with the one-dimensional approximation is a useful baseline, even for more complex surfaces.

The two different versions of the two-dimensional Navier-Stokes equations show that there are multiple ways of approaching the problem of approximating the velocity of the system of fluid flow through a circular pipe. The gauge method uses an auxiliary field to approximate the pressure via a gauge transformation and does not consider external forces affecting the system, whereas the two-dimensional formulation considering external forces approximates the pressure at each step via FFT and then approximates the velocities. In both cases the initialization of the system is what causes the differences, which include the initial velocities, initial pressure and, for the case considering forces, the external forces. Aside from this the various formulations arrive at similar approximations and neither formulation have singularities or irregularities present in the domain. The formulations presented preserve the boundary conditions well and we are able to minimize subtleties related to the inner wall of a circular pipe.

Appendix A. Fast Fourier Transform

The general Fourier Transform of an integrable function $f : R - - > C$ is an integral transform given by the following formula:

$$\mathscr{F}\{f(t)\} = \hat{f}(k) = \int_{-\infty}^{\infty} e^{-2\pi ikt} f(t)\, dx \tag{A.1}$$

While the inverse Fourier Transform is defined as follows:

$$\mathscr{F}^{-1}\{f(t)\} = f(t) = \int_{-\infty}^{\infty} e^{-2\pi ikt} \hat{f}(k)\, dx \tag{A.2}$$

As described in [SM19], the function above takes the function $f(t)$ as input and outputs the function $\hat{f}(k)$ where t is time and k is frequency. The functions f and $\hat{f}$ are called a Fourier Transform pair. There are different ways to define the Fourier Transform, and we will look closer at the Fast Fourier Transform (FFT) which is what we used in solving the Poisson equation for pressure in the two-dimensional Navier-Stokes formulation with forces.

The best place to start to get a good understanding of FFT is to first look at the Discrete Fourier Transform (DFT). After all, FFT refers to an efficient implementation of DFT for highly composite lengths N [SM08]. We will give an example of how to compute FFT and suggest that the built-in function fft in Matlab used in section 3.2 follows a similar procedure. Whenever the length N of the DFT can be described as a product of smaller integers, we can get a decomposition known as a mixed radix Cooley-Tukey FFT algorithm [SM08]. Then the DFT is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, k = 0, 1, 2, ..., N-1 \tag{A.3}$$

Where x(n) is the input signal amplitude at time n, and W_N is given as:

$$W_N \triangleq e^{-j\frac{2\pi}{N}} \tag{A.4}$$

Which is the primitive Nth root of unity. This also means that $W_N^N = 1$. Now, when N is even, we can split the DFT summation into two separate summations over the odd and even indices of the input signal as follows:

$$\begin{gathered} \mathrm{X}(\omega_k) \triangleq DFT_{N,k}x \triangleq \textstyle\sum_{n=0}^{N-1} x(n)e^{-j\omega_k nT}, \omega_k \triangleq \frac{2\pi k}{NT} = \sum_{n=0}^{N-2} x(n)e^{-j\omega_k nT} + \\ \textstyle\sum_{n=0}^{N-1} x(n)e^{-j\omega_k nT} = \sum_{n=0}^{\frac{N}{2}-1} x(2n)e^{-j2\pi\frac{k}{\frac{N}{2}}n} + e^{-j2\pi\frac{k}{N}} \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)e^{-j2\pi\frac{k}{\frac{N}{2}}n} = \\ \textstyle\sum_{n=0}^{\frac{N}{2}-1} x_e(n)W_{\frac{N}{2}}^{kn} + e^{-j2\pi\frac{k}{N}} \sum_{n=0}^{\frac{N}{2}-1} x_o(n)W_{\frac{N}{2}}^{kn} \triangleq \\ DFT_{\frac{N}{2},k}DOWNSAMPLE_2(x) + W_N^k \cdot DFT_{\frac{N}{2},k}DOWNSAMPLE_2[SHIFT_1(x)] \end{gathered} \tag{A.5}$$

Where $x_e(n) \triangleq x(2n)$ and $x_o(n) \triangleq x(2n+1)$ denote the even and odd indexed samples from x. Therefore, the DFT of length N is computable using two DFTs of length $\frac{N}{2}$.

The built-in fft function in MATLAB is widely used since it is adaptive and therefore chooses the best algorithm available for the desired transform size [SM08].

Appendix B. MATLAB code

For one-dimensional Navier-Stokes the MATLAB code used is as follows:

```
nx = 50;
nt = 50;
xmax=2*pi;
u = zeros(nx, nt);
v = zeros(nx, nt);
x = 0:xmax/(nx-1):xmax;
tmax = 1;
t=0:tmax/(nt-1):tmax;
a = 2;
b = 4;
c = 0.1;
dt = 1/nt;
dx = 2*pi/nx;
u(1,:)=sin(x(:));
v(1,:) = sin(x(:));
u(:,1)=1/a * exp(-b)-1/a * exp(-a*t(:)-b);
v(:,1) = 0;
v(:,nx) = 0;
u(:,nx)=1/a * exp(-b)-1/a * exp(-a*t(:)-b);
for it = 2:nt-1
    for ix = 2:nx-1
        u(ix, it) = u(ix-1,it) - u(ix-1,it)*dt*(u(ix-1,it+1)-u(ix-1,it-1))/(2*dx) ...
        +c*dt*((u(ix-1,it+1) -2*u(ix-1, it)+u(ix-1,it-1))/(dx.^2));
        v(ix, it) = v(ix-1,it) - v(ix-1,it)*dt*(v(ix-1,it+1)-v(ix-1,it-1))/(2*dx) ...
        +c*dt*((v(ix-1,it+1)-2*v(ix-1, it)+v(ix-1,it-1))/(dx.^2));
    end
end

X = reshape(x,50,[]);
Y = reshape(t,50,[]);

figure
subplot(1,2,1) ; surf(X,Y,u)
xlabel('position')
ylabel('time')
zlabel('velocity')
```

```
finalerror = zeros(40, 1);
for i = 1:nt
    error = abs(u(:, i) - v(:,i));
    sum(error);
    if i <= 40
        finalerror(i) = sum(error);
    end
end
q = 1:40;

figure
loglog(q, finalerror)
xlabel('time')
ylabel('error')
```

For the two-dimensional case using gauge formulation the following code was used. Keep in mind that the pressure was changed for the different systems presented in section 3.1:

```
nx = 150;
ny = 150;
nt = 10;
xmax=1;
u = zeros(nx, ny);
test = u;
w = zeros(nx, ny);
a = zeros(nx, ny);
v = zeros(nx, ny);
aa = zeros(nx, ny);
bb = zeros(nx, ny);
x = -1:xmax*2/(nx-1):xmax;
h = xmax*2/(nx-1);
ymax = 1;
tmax = 1;
y=-1:ymax*2/(ny-1):ymax;
Re = 40000;
p = zeros(nx ,ny);
pvar = zeros(nx, ny);
dt = 1/nt;
dh = 0.01/nt;
dx = 2*pi/nx;
for it = 1:nt-1
    for iy = 1:ny
        for ix = 1:nx
            if ((x(ix))^2 + (y(iy)^2)) == 1
                if it ==1
                    p(ix, iy) = -cos((pi/2)*((x(ix))^2+(y(iy))^2));
                    pvar = p;
                    if (((x(ix+1))^2 + (y(iy)^2)) > 1)
                        pvar(ix+1, iy) = pvar(ix, iy);
                    end
                    if (((x(ix))^2 + (y(iy+1)^2)) > 1)
                        pvar(ix, iy+1) = pvar(ix, iy);
                    end
                    u(ix, iy) =  cos((pi/2)*((x(ix))^2+(y(iy))^2));
                    a(ix, iy) = u(ix, iy)+(pvar(ix+1, iy)+pvar(ix, iy+1)-2*p(ix, iy));
                end
                if it >1
                    w = u;
                    pvar = p;
                    aa =a;
                   if (((x(ix+1))^2 + (y(iy)^2)) > 1)
                        aa(ix+1, iy) =  aa(ix, iy);
                        pvar(ix+1, iy) = pvar(ix, iy);
                        w(ix+1, iy) = w(ix, iy);
                   end
                    if (((x(ix))^2 + (y(iy+1)^2)) > 1)
                        aa(ix, iy+1) =  aa(ix, iy);
                        pvar(ix, iy+1) = pvar(ix, iy);
                        w(ix, iy+1) = w(ix, iy);
                    end
                    if (((x(ix-1))^2 + (y(iy)^2)) > 1)
```

```
                    aa(ix-1, iy) =  aa(ix, iy);
                    pvar(ix-1, iy) = pvar(ix, iy);
                    w(ix-1, iy) = w(ix, iy);
                end
                if (((x(ix))^2 + (y(iy-1)^2)) > 1)
                    aa(ix, iy-1) =  aa(ix, iy);
                    pvar(ix, iy-1) = pvar(ix, iy);
                    w(ix, iy-1) = w(ix, iy);
                end
                a(ix ,iy) = (1/Re)*dt*((aa(ix-1, iy)+aa(ix+1, iy)+aa(ix, iy-1)+aa(ix, iy+1
                p(ix, iy) = ((pvar(ix-1, iy)+pvar(ix+1, iy)+pvar(ix, iy+1)+pvar(ix, iy-1))
                u(ix, iy) = a(ix, iy) - (pvar(ix+1, iy)+pvar(ix, iy+1)-2*p(ix, iy));
            end
        end
        if ((x(ix))^2 + (y(iy)^2)) > 1
            u(ix, iy) = 0;
            p(ix, iy) = 0;
            a(ix, iy) = 0;
        end
        if ((x(ix))^2 + (y(iy)^2)) < 1
            if it ==1
                p(ix, iy) = -cos((pi/2)*((x(ix))^2+(y(iy))^2));
                pvar = p;
                if (((x(ix+1))^2 + (y(iy)^2)) > 1)
                    pvar(ix+1, iy) = pvar(ix, iy);
                end
                if (((x(ix))^2 + (y(iy+1)^2)) > 1)
                    pvar(ix, iy+1) = pvar(ix, iy);
                end
                u(ix, iy) = cos((pi/2)*((x(ix))^2+(y(iy))^2));
                a(ix, iy) = u(ix, iy)+(pvar(ix+1, iy)+pvar(ix, iy+1)-2*p(ix, iy));
            end
            if it >1
                w = u;
                pvar = p;
                aa=a;
                if (((x(ix+1))^2 + (y(iy)^2)) > 1)
                    aa(ix+1, iy) =  aa(ix, iy);
                    pvar(ix+1, iy) = pvar(ix, iy);
                    w(ix+1, iy) = w(ix, iy);
                end
                if (((x(ix))^2 + (y(iy+1)^2)) > 1)
                    aa(ix, iy+1) =  aa(ix, iy);
                    pvar(ix, iy+1) = pvar(ix, iy);
                    w(ix, iy+1) = w(ix, iy);
                end
                if (((x(ix-1))^2 + (y(iy)^2)) > 1)
                    aa(ix-1, iy) =  aa(ix, iy);
                    pvar(ix-1, iy) = pvar(ix, iy);
                    w(ix-1, iy) = w(ix, iy);
                end
                if (((x(ix))^2 + (y(iy-1)^2)) > 1)
                    aa(ix, iy-1) =  aa(ix, iy);
                    pvar(ix, iy-1) = pvar(ix, iy);
                    w(ix, iy-1) = w(ix, iy);
                end
                a(ix ,iy) = (1/Re)*dt*((aa(ix-1, iy)+aa(ix+1, iy)+aa(ix, iy-1)+aa(ix, iy+1
                p(ix, iy) = ((pvar(ix-1, iy)+pvar(ix+1, iy)+pvar(ix, iy+1)+pvar(ix, iy-1))
                u(ix, iy) = a(ix, iy) - (pvar(ix+1, iy)+pvar(ix, iy+1)-2*p(ix, iy));
            end
        end
    end
end
```

```
X = reshape(x,nx,[]);
Y = reshape(y,ny,[]);

levels = 0:0.1:10;
figure
subplot(1,1,1) ; contour(X,Y,u, levels, 'k')
xlabel('x')
ylabel('y')
```

For the two-dimensional case with external forces this is the MATLAB code we used. Keep in mind that the pressure and forces were changed for the different systems presented in section 3.2:

```
nx = 100;
ny = 100;
nt = 10;
xmax=1;
u = zeros(nx, ny);
uhat = zeros(nx, ny);
phat = zeros(nx, ny);
x = -1:xmax*2/(nx-1):xmax;
uu = zeros(nx, ny);
ymax = 1;
tmax = 1;
y=-1:ymax*2/(ny-1):ymax;
p = zeros(nx ,ny);
phi = zeros(nx, ny);
h = xmax*2/(nx-1);
dt = 1/nt;
dx = 2*pi/nx;
Re = 40000;
rho = 1;
for it = 1:nt
    for iy = 1:ny
        for ix = 1:nx
            if ((x(ix))^2 + (y(iy)^2)) == 1
                if it == 1
                    u(ix, iy) = cos((pi/2)*((x(ix))^2+(y(iy))^2));
                end
                if it > 1
                    uu = u;
                    phi = p;
                    if (((x(ix+1))^2 + (y(iy)^2)) > 1)
                        uu(ix+1, iy) =  uu(ix, iy);
                        phi(ix+1, iy) = phi(ix, iy);
                    end
                    if (((x(ix))^2 + (y(iy+1)^2)) > 1)
                        uu(ix, iy+1) =  uu(ix, iy);
                        phi(ix, iy+1) =  phi(ix, iy);
                    end
                    if (((x(ix-1))^2 + (y(iy)^2)) > 1)
                        uu(ix-1, iy) =  uu(ix, iy);
                        phi(ix-1, iy) =  phi(ix, iy);
                    end
                    if (((x(ix))^2 + (y(iy-1)^2)) > 1)
                        uu(ix, iy-1) =  uu(ix, iy);
                        phi(ix, iy-1) =  phi(ix, iy);
                    end
                    u(ix, iy) = uu(ix, iy) - dt*uu(ix, iy)*(uu(ix+1,iy)+uu(ix, iy+1)-2*uu(ix,
                    temp = (uu(ix+1, iy)-2*u(ix, iy)+uu(ix, iy+1));
                    phat(ix, iy) = fft((1/dt)*(temp));
                    uhat(ix, iy) =  (phat(ix, iy))/(ix^2 + iy^2);
                    temp1 = uhat(ix+1, iy);
                    temp2 = uhat(ix, iy+1);
                    p(ix, iy) = ifft((rho/dt)*(temp1-2*uhat(ix, iy)+temp2));
                    u(ix, iy) = u(ix, iy)- (dt/rho)* ((phi(ix+1, iy)-2*p(ix, iy)+phi(ix, iy+1)
```

```
            end
        end
        if ((x(ix))^2 + (y(iy)^2)) > 1
            u(ix, iy) = 0;
        end
        if ((x(ix))^2 + (y(iy)^2)) < 1
            if it == 1
                u(ix, iy) = cos((pi/2)*((x(ix))^2+(y(iy))^2));
            end
            if it > 1
                uu = u;
                phi = p;
                if (((x(ix+1))^2 + (y(iy)^2)) > 1)
                    uu(ix+1, iy) =  uu(ix, iy);
                    phi(ix+1, iy) = phi(ix, iy);
                end
                if (((x(ix))^2 + (y(iy+1)^2)) > 1)
                    uu(ix, iy+1) =  uu(ix, iy);
                    phi(ix, iy+1) =  phi(ix, iy);
                end
                if (((x(ix-1))^2 + (y(iy)^2)) > 1)
                    uu(ix-1, iy) =  uu(ix, iy);
                    phi(ix-1, iy) =  phi(ix, iy);
                end
                if (((x(ix))^2 + (y(iy-1)^2)) > 1)
                    uu(ix, iy-1) =  uu(ix, iy);
                    phi(ix, iy-1) =  phi(ix, iy);
                end
                u(ix, iy) = uu(ix, iy) - dt*uu(ix, iy)*((uu(ix+1,iy)+uu(ix, iy+1)-2*uu(ix,
                temp = (uu(ix+1, iy)-2*u(ix, iy)+uu(ix, iy+1));
                phat(ix, iy) = fft((1/dt)*(temp));
                uhat(ix, iy) =  (phat(ix, iy))/(ix^2 + iy^2);
                temp1 = uhat(ix+1, iy);
                temp2 = uhat(ix, iy+1);
                p(ix, iy) = ifft((rho/dt)*(temp1-2*uhat(ix, iy)+temp2));
                u(ix, iy) = u(ix, iy)- (dt/rho)* ((phi(ix+1, iy)-2*p(ix, iy)+phi(ix, iy+1)
            end
        end
    end
end
```

```
X = reshape(x,nx,[]);
Y = reshape(y,ny,[]);

levels = 0:0.1:10;
figure
subplot(1,1,1) ; contour(X,Y,u, levels, 'k')
xlabel('x')
ylabel('y')
```

Bibliography

[AC17] N. Arada and P. Correia, *Second-grade fluid in curved pipes*, July 2017.

[AD99] B. Alessandrini and G. Delhommeau *A fully coupled Navier–Stokes solver for calculation of turbulent incompressible free surface flow past a ship hull*, International journal for numerical methods in fluids 29.2 (1999): 125-142.

[BB09] H. Bae and L. Brandolese *On the effect of external forces on incompressible fluid motions at large distances*, April 2009.

[BCM01] D. L. Brown, R. Cortez and M. L. Minion, *Accurate Projection Methods for the Incompressible Navier-Stokes Equations*, Journal of Computational Physics 168, p 464-499, 2001.

[BGWX22] J. Bang, C. Gui, Y. Wang and C. Xie, *Liouville-type Theorems for Steady Solutions to the Navier-Stokes System in a Slab*, May 2022.

[BSDHO21] J.F.H. Buist, B. Sanderse, S. Dubinkina, R.A.W.M. Henkes and C.W. Oosterlee, *Energy-conserving formulation of the two-fluid model for incompressible two-phase flow in channels and pipes*, December 2021.

[BW59] W. Bauer, G. D. Westfall, *University Physics*, Figure 13.42, 1959.

[BW78] R. M. Beam and R. F. Warming, *An implicit factored scheme for the compressible Navier-Stokes equations*, AIAA journal 16.4 (1978): 393-402.

[EL03] Weinan E and J. Liu, *Gauge Method for Viscous Incompressible Flows*, Comm. Math. Sci. Vol 1, No. 2 pp 317-332, 2003.

[GR00] G. Grätzer, *Math into LaTeX: an introduction to LaTeX and AMS-LaTeX, Third Edition*, Library of Congress, Boston, 2000.

[LA12] H. P. Langtangen, *Numerical methods for the Navier-Stokes equations*, Center for Biomedical Compuing, Simula Research Laboratory, Department of Informatics, University of Oslo, 2012.

[LD22] B. Lopez-Doriga and S. T. M. Dawson, *Resolvent analysis of laminar and turbulent duct flows*, May 2022.

[LL59] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, Course of theoretical physics, 1959.

[LRPM22] D. Liu, F. Rist, H. Pottmann, D. Michels, *UrbanFlow: Designing Comfortable Outdoor Areas*, April 2022.

[LS19] LSU Graduate School, *Formatting Electronic Theses and Dissertations*, October 2019.

[MA22] Mathworks, *Create Surface Plot*, R2022a, May 2022.

[MA22] Mathworks, *Contours of a Function*, R2022a, May 2022.

[MFG04] F. Mittelbach and M. Goossens, *The LaTeX Companion, Second Edition*, Addison-Wesley, Boston, 2004.

[MMABW20] G. Muralova, T. Martynova, E. Andreeva, V. Bavin and Z. Wang, *Numerical Solution of the Navier-Stokes Equations Using Multigrid Methods with HSS-Based and STS-Based Smoothers*, 2020.

[MMN21] S. L. Mason, J. C. Meyer and D. J. Needham, *The Development of a Wax Layer on the Interior Wall of a Circular Pipe Transporting Heated Oil - The Effects of Temperature Dependent Wax Conductivity*, April 2021.

[PA17] A. Paglietti, *The laminar-to-turbulent transition in viscous fluid flow*, March 2017.

[RH21] C. A. Rosales-Alcantar and G. Hernandez-Duenas, *A new two-dimensional blood flow model with arbitrary cross sections*, August 2021.

[SB05] R. E. Sonntag and C. Borgnakke, *Introduction to Engineering Thermodynamics, Second Edition*, p 468-473, 2005.

[SM08] J. O. Smith III, *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications, Second Edition*, p 197-205, 2008.

[SM19] G. Smith, *The Fast Fourier Transform and its Applications*, 2019.

[SO13] T. Sochi, *One-Dimensional Navier-Stokes Finite Element Flow Model*, 2013

[ST86] G. Strang, *Introduction to Applied Mathematics*, p 235-236, 1986.

[TB20] P. M. Tripathi and S. Basu, *Effect of wall roughness on heat transfer in supercritical water flow*, December 2020

[TO20] A. Toutant, *Numerical simulations of unsteady viscous incompressible flows using general pressure equation*, May 2020.

[VIL22] Q. Vila, *Time-Asymptotic Study of a Viscous Axisymmetric Fluid without Swirl*, May 2022.

[WS12] J. Wiley and Sons, *Continuum Fluid Mechanic and the Navier-Stokes Equations*, Understanding Aerodynamics: Arguing from the Real Physics, p 13-78, 2012.

[XHBC21] J. Xie, J. He, Y. Bao, X. Chen, *A low-communication-overhead parallel method for the 3D incompressible Navier-Stokes equations*, April 2021.

[ZE19] P. Zeman, *Discrete and Fast Fourier Transform Made Clear*, August 2019.

Vita

Jon Nerdal was born in Bergen, Norway in February 1996. He earned a Bachelor of Arts in Mathematics from the University of Minnesota in 2020 while being a student athlete on the track team. In August of 2020 he came to Louisiana State University to start graduate studies in mathematics, and he was also a student athlete on the track team at LSU while completing his work for his Master's Degree in Mathematics. He plans to receive his Master's from LSU in August 2022.