Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Bob Bao April 6, 2019

Surrogate models for incompressible fluid dynamics in periodic regime

by

Bob Bao

Alessandro Veneziani
Adviser

Department of Mathematics

Alessandro Veneziani

Adviser

Manuela Manetta

Committee Member

Steve La Fleur

Committee Member

2019

Surrogate models for incompressible fluid dynamics in periodic regime

By

Bob Bao

Alessandro Veneziani

Adviser

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics

2019

Abstract

Surrogate models for incompressible fluid dynamics in periodic regime
By Bob Bao

Computational fluid dynamics (CFD) plays an important role in modeling the system of Left Ventricle Assist Device (LVAD), which is now a main solution for the patients who reach to an end-stage heart failure. However, because the post-surgery conditions create an abnormal hemodynamics that may eventually lead to long-term complications and diseases in aorta. CFD can be used to understand the interplay between post-op morphology, hemodynamics and clinical outcomes. However, a main fallback of using CFD is the high computational cost. In order to reach to accurate solutions with an inexpensive cost, a surrogate model for the unsteady Navier-Stokes equation is preferred. Acknowledging that heart has a periodic behavior in beats, we try to use the solution for steady Navier-Stokes equation to approximate the time-average solution for unsteady Navier-Stokes equation in periodic regime. Because of the existence of the non-linear term in the Navier-Stokes equation, this approximation will present some differences between those two solutions. By using FreeFem++ to run the numerical simulations on different geometries for those two problems, this paper will discuss how the differences between those two solutions will be affected by various of factors including the amplitude of boundary conditions, number of time steps at which the unsteady problem is solved, the quality of meshes and different geometries. From the results obtained from the numerical tests, it is concluded that the geometry will have a dominated effect on the differences between those two solutions. When the geometry is regular and indicates a non-linear term approaching to 0, the amplitude of the boundary condition will have a polynomial-like relationship with that difference. The quality of messes and number time steps will only affect the computational cost but the differences between those two solutions.

Surrogate models for incompressible fluid dynamics in periodic regime

By

Bob Bao

Alessandro Veneziani

Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics

2019

Contents

1 Introduction **9**
- 1.1 Motivations . . . 9
- 1.2 Outline . . . 11

2 The Stokes and the Navier-Stokes Equations **13**
- 2.1 The Stokes flow problem . . . 13
- 2.2 The Navier-Stokes problem . . . 15
- 2.3 Numerical approximation of the (Navier)-Stokes equations . . . 15

3 Surrogates for periodic equations **17**
- 3.1 The linear case (Stokes) . . . 17
- 3.2 The nonlinear case (Navier-Stokes) . . . 18

4 The numerical code **21**
- 4.1 The FreeFem++ environment . . . 21
- 4.2 Numerical test 1: flow in a cylinder . . . 21
  - 4.2.1 The meshes and boundary conditions . . . 21
  - 4.2.2 The algorithm to solve the steady problem . . . 22
  - 4.2.3 The algorithm to solve the unsteady problem . . . 24
  - 4.2.4 Test the code's correctness in cylinder . . . 24
  - 4.2.5 Variables set in the code for testing . . . 24
- 4.3 Numerical test 2: flow in a room . . . 24
  - 4.3.1 The meshes and boundary conditions . . . 24
  - 4.3.2 The algorithm to solve the problem . . . 25

5 Numerical results **27**
- 5.1 Test case 1: Flow in a cylinder . . . 27
  - 5.1.1 The linear case: code testing . . . 27
  - 5.1.2 The nonlinear case . . . 27
- 5.2 Test case 2: Flow in a room (Navier-Stokes Problem) . . . 33
  - 5.2.1 Changing the quality of the meshes . . . 33
  - 5.2.2 Changing the inflow factor of the boundary condition . . . 34
- 5.3 Discussion . . . 35
  - 5.3.1 For the nonlinear term in the equation . . . 35
  - 5.3.2 For the irregular geometry . . . 35

5.3.3 Future directions 36

References **39**

Chapter 1

Introduction

1.1 Motivations

In the field of the medicine, computational modeling is used pretty often in the case of helping doctors make a better decision and prediction for the diagnosis of the patients. The LVAD surgery is a good example. LVAD, defined as the *Left Ventricular Assist Device*, is a pump that is used to implant in the patients who have reached the end-stage heart failure [2]. As the figure shown below (figure 1.1), this surgery is giving a "new heart" to those people with cardiovascular diseases such as poor cardiac ejection fraction [3]. With this "new heart", it can than pump the fresh blood through a hole made on the aorta, which should be originally pumped through the connection between the aorta and the heart. However, the post-surgery conditions create an abnormal hemodynamics that eventually may lead to long-term complications and diseases in the aorta. Computational models may have an important role in understanding the interplay between post-op morphology, hemodynamics and clinical outcomes.

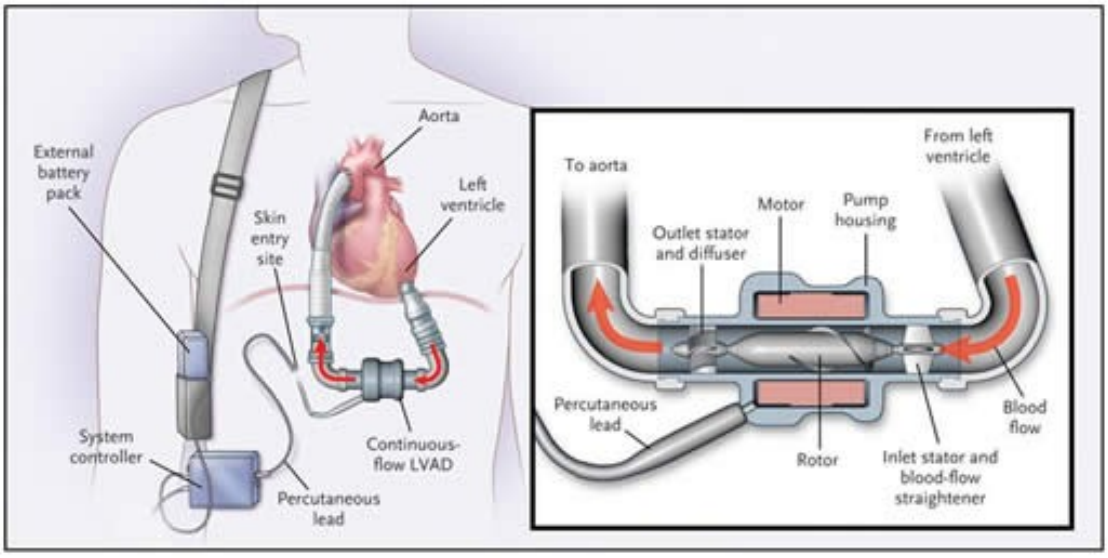


Figure 1.1: This is the demonstration of LVAD presenting as a "new heart" in man's body. [2]

In order to make better prediction and have a better understanding of the effect of implanting LVAD in the patient's body and the ideal position of the injection the pipe of the pump on the aorta, we need to introduce the computational fluid dynamics(CFD) to simulating the flow of the blood inside the vessels. With those simulation, we can predict how a small change in the position, direction, rate or the strength of the injection will affect patients in long term and make a better decision for the LVAD surgery based on that to make the effect of LVAD last longer. All these features, including position, direction, rate or strength can be characterized by the change of boundary conditions in the CFD model. To reach to the goal to investigate the effect of those details of the surgery in the CFD model, we're just researching on how the change in boundary conditions can affect the results of the CFD model. In this way, the cardiovascular and vessels system with LVAD can be simulating by a CFD model with different boundary conditions.

The efficiency of solving the simulation in CFD is always important because of the time cost for solving the problem and the complexity of the problem. Surrogate models are more interested on using simpler models to reach to the goals or effects we needed for the problem. For example, while doing numerical approximation, interpolation of the functions is always a important way of finding the trends and some certain data points of the function. Sometimes we we'll only know some of the data points of a function but we need some for data from that function. Interpolating the data points with polynomials will make it easier to find the data for the function. With this, we successfully avoided extrapolating the function directly, but rather focused on finding the rest of the data points on the function. Another example appears when solving the system above, it is really hard directly on the shape of the human vessels itself since the vessel of human being has an irregular shape and it is really hard to solve equation directly over the shape of the vessels itself. The surrogate models is needed here for the problems to be solved more efficiently. To reduce the computational costs, one possible way consists of simplifying the geometry, discarding a precise 3D

geometrical modeling in favor of a pure 1D [1].

There is specific surrogate in the numerical approach which is approaching a time dependent problem with periodic boundary conditions (an unsteady periodic problem) with a time independent problem (a steady problem) in CFD. Admittedly the time dependent problem is detailed and informative since it will provide a solution after a defined time period with the full detailed information of the data of the simulation at each time points. This is helpful sometimes when you need to find the prediction of a patients data, but there are some drawbacks of numerically solving the time dependent problem. Since there are tons of data for a single patients (for each time point), it is hard to compare the data across the patients which is also an important component of the medical research. However, the most significant drawbacks of the transient problem is that they are computational expensive [3]. Depending on the computational resources and the simulated time interval set for the simulation, the time of solving a unsteady problem can varies from hours to days[3]. Since usually accurate simulations are needed for medical solutions, it means in most of the simulations, large computational resources and small time interval is needed which will lead to an high computational cost. That's why it becomes so important to use the solution of the steady problem to approach the unsteady problem to reduce the computational cost.

Replacing an unsteady problem with a steady one is possible here because we can reasonably assume that the circulation is time-periodic. As we will see, if you properly exploit this feature, we can approximate the unsteady probloem with a steady one. *The main purpose of this work is to test to which extent, and under what flow conditions this simplification may be considered acceptable.*

1.2 Outline

We first discuss how the surrogate from an unsteady problem to a steady one works in some basic equations of fluid dynamics such as the Stokes problem and the Navier-Stokes problem. Then we introduce the numerical environment that we're using to solve this problem together with the code and the 3D models that we are applying to. Finally we'll show the results of running the simulations, how the time-averaged solutions in the unsteady Navier-Stokes equations are different from the steady Navier-Stokes equations and how the non-linear part in the Navier-Stokes equations will affect this surrogate from the unsteady problem to the steady one.

Specifically, in Chapter 2 we present the basic mathematical models for incompressible fluid dynamics, used in cardiovascular applications. In Chapter 3 we introduce the rationale of our surrogate modeling when we replace the unsteady problem with a time average, resulting in a steady system. This is invariably much faster to solve. In Chapter 4 we illustrate the numerical approximation of the problems via the FreeFem++ environment. Results are discussed in Chapter 5.

Chapter 2

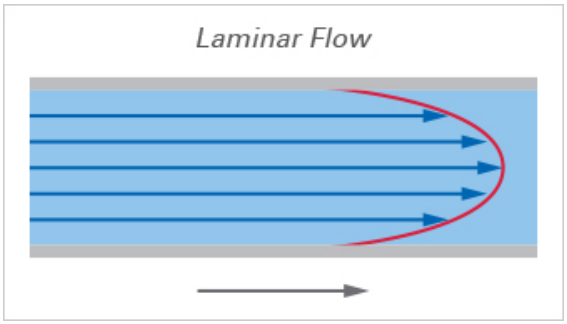
The Stokes and the Navier-Stokes Equations

2.1 The Stokes flow problem

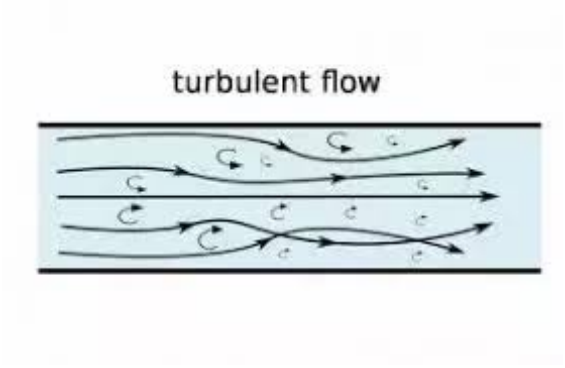
The basic model for incompressible fluids is given by four partial differential equations known as "Navier-Stokes systems". These equations follow from the application of conservation of momentum and conservation of mass to a fluid. Specifically, we will assume that the liquid at hand is "Newtonian" (so that the stress tensor is proportional to the strain rate through a constant called "viscosity"). As we will see, the system is non-linear and it is generally impossible to find a solution by analytical methods. Numerical approximations are necessary. However, under some circumstances, the nonlinear term can be dropped, resorting to the so-called "Stokes equations". We start here introducing this problem.

Since the Navier-Stokes equation has not been solved in general case, the only possible ways to get a solution for the problem is to find an analytical solution for a simpler geometry (for example, getting a solution in 1D) or find an Stokes analytical solution for a simpler equation that has some limit in it [7]. Stokes equation for the stokes flow problem and Laplace equations are the examples of solving Navier-Stokes problem with those simpler equations. We'll give an detailed explanation of the Stokes flow problem here since this is a problem that can be perfectly applied to the approach introduced in the last chapter: surrogate the time-averaged solution of an unsteady problem with a steady problem.

The Stokes flow problem, also called the *creeping flow problem*, works under low Reynolds number. The Reynolds number is an important factor that illustrates the flow regime. When the Reynolds number is low, fluid mainly behaves as laminar flow (shown by the graph 2.1a) and when the Reynolds number is high, it suggests that disturbed or even turbulent flow (shown by the graph 2.1b) will dominates the behavior under that environment. Strictly speaking, it requires Reynolds number to be far less than 1. However, sometimes it is considered a good approximation under more general conditions.



(a) This is the graph for the laminar flow. We assume the layer like behaviour of the fluid when the Reynolds number is small.
[6]



(b) This is the graph for the turbulent flow. We assume this behaviour when the Reynolds number is large.
[9]

Figure 2.1: This is the graphs for two different flow types.

The Stokes equation with incompressible fluid is stated below:

$$\begin{cases} \rho \dfrac{\partial \boldsymbol{u}}{\partial t} - \mu \Delta \boldsymbol{u} + \nabla p = \mathbf{f} \\ \nabla \cdot \boldsymbol{u} = 0 \end{cases} \tag{2.1}$$

For the first equation, ρ stands for the density of the fluid, μ stands for the kinematic viscosity of the fluid and this will varies among different fluids. $\boldsymbol{u}$ stands for the three dimensional velocity vector of the flow of the fluid. p stands for the pressure of the fluid and $\mathbf{f}$ stands for the non-homogeneous forcing term for the stokes equation. The second equation, which states the divergence of the fluid is 0, represents that the fluid is incompressible.

There is also a steady version for the Stokes equation:

$$\begin{cases} -\mu \Delta \boldsymbol{u} + \nabla p = \mathbf{f} \\ \nabla \cdot \boldsymbol{u} = 0 \end{cases} \tag{2.2}$$

In the applications, the space variables (x_1, x_2, x_3) belongs to a limited part of space denoted by Ω and on a time interval [0,T]. For this reason, in the applications, we need boundary conditions on $\partial\Omega$ and an initial condition. Possible boundary conditions prescribe the velocity u on $\partial\Omega$ and the initial velocity $u(x, 0) = u_0(x)$. Other boundary conditions can be considered as well. For example the natural (Neumann) conditions read $pn - \mu \nabla u \cdot n$ where n is the outward normal unit vector to $\partial\Omega$. Finally, if we assume that the problem is periodic in time, we may consider $u(x, 0) = u(x, T)$. While one can exploit this feature directly in the solution (both analytical and numerical), here we will limit to apply this assumption in surrogating the original unsteady problem with a steady one.

2.2 The Navier-Stokes problem

The Navier-Stokes equation is a generalization of the equation provided by Lenohard Euler made in the 18th century which describes the behavior of the frictionless and incompressible fluid. In 1821, French engineer Claude-Louis Navier, introduced the element of viscosity for the more realistic condition of the fluid. With the improvement made by Sir George Gabriel Stokes in the middle of the 19th century, some solution of the Navier-Stokes equation has been unveiled such as the case of simple two dimensional flow. However, the general 3D Navier-Stokes equation has been proven to be intractable in general and can only be approximated by numerical approach [5].

The equation of the Navier Stokes problem is shown below:

$$\begin{cases} \rho \dfrac{\partial \boldsymbol{u}}{\partial t} + \rho \left(\boldsymbol{u} \cdot \nabla \right) \boldsymbol{u} - \mu \Delta \boldsymbol{u} + \nabla p = \mathbf{f} \\ \\ \nabla \cdot \boldsymbol{u} = 0 \end{cases} \tag{2.3}$$

Here, all the variables in the equation has the same meaning as the Stokes equation. The only difference of this equation compared to the Stokes equation is that it contains the non-linear term stated as $\rho \left(\boldsymbol{u} \cdot \nabla \right) \boldsymbol{u}$.

There is also a steady version (time independent) of the equation which is presented below. When the time derivative is set to 0:

$$\begin{cases} \rho \left(\boldsymbol{u} \cdot \nabla \right) \boldsymbol{u} - \mu \Delta \boldsymbol{u} + \nabla p = \mathbf{f} \\ \\ \nabla \cdot \boldsymbol{u} = 0 \end{cases} \tag{2.4}$$

Also in this case, the problem must be completed by boundary and initial conditions. As for the Stokes problem, if we assume the problem to be periodic in time, we may introduce some simplifications that will be addressed in the next Chapter.

This equation has all the variable the same meaning as the full version of the Navier-Stokes equation but the only thing different here is that the time dependent term is eliminated. Unlike the Stokes equations, there is no specific physical meaning for the steady and it is just a numerical trick for people to solve on the path of solving the full Navier-Stokes equation. However, it can be a good way of surrogating the full Navier-Stokes equation in some way.

2.3 Numerical approximation of the (Navier)-Stokes equations

Since the Navier-Stokes Equation at this moment has not been solved in the general 3D cases, a way of numerical approaching is needed to find a numerical solution to the equation while doing the simulation. The method of numerical approximation applied in this research is the method of space discretization [8]. This is a method based on the replacement of the derivatives by incremental quotients after descretisizing the onto a suitable grid of points [8]. These points are actually where the problems are solved. These points are usually chosen along the Cartesian directions [8], which, in this research, is represented by the vertices and the midpoint of the sides of the tetrahedrons in the mesh. These points are actually the spaces for the numerical solution, that is, where the solution is actually solved [8]. In this research, we are using FreeFem++, a popular 3D partial

differential equation solver across the world in which the physics for solving incompressible Navier-Stokes equation with space discretization has already been built [4].

Chapter 3

Surrogating the unsteady periodic equations with steady problems

In many medical applications, doctors are not interested in the instantaneous quantities (velocity and pressure) but to their time average. This is true - for instance - when looking at the stress exerted by blood on the wall, that is considered a main factor for atheorsclerosis or when evaluating the pressure drop induced by a vascular occlusion in coronary diseases. These time averages are generally computed by taking the average of the unsteady solution. However it is reasonable to ask the question: is it possible to compute directly the time average by solving a steady problem? What if we take the time average of the Stokes and Navier-Stokes equations and then we solve the problem we obtain? In this Chapter we show how the surrogate steady problems look like and what is the approximation introduced.

3.1 The linear case (Stokes)

Between the two stokes equations in the linear(Stokes) cases, we can find there are certain relationship between the equation 2.1 and 2.2 when the problem is solved under periodic conditions.

Here is the deduction:

Denote by ΔT the period of our problem and compute the time average of the unsteady Stokes equations over one period. We obtain (REMOVE p for the SPACE VARIABLES, IT IS MESSING UP WITH p=pressure).

$$\frac{1}{\Delta T}\int_t^{t+\Delta T}(\rho\frac{\partial \boldsymbol{u}}{\partial t}-\mu\Delta\boldsymbol{u}+\nabla p=\mathbf{f}),$$

$$\frac{1}{\Delta T}\int_t^{t+\Delta T}\rho\frac{\partial \boldsymbol{u}}{\partial t}-\frac{1}{\Delta T}\int_t^{t+\Delta T}\mu\Delta\boldsymbol{u}+\frac{1}{\Delta T}\int_t^{t+\Delta T}\nabla p=\frac{1}{\Delta T}\int_t^{t+\Delta T}\mathbf{f},$$

Because of the linearity of all the terms, we have, (3.1)

$$\frac{\rho}{\Delta T}[\boldsymbol{u}(\boldsymbol{x},t+\Delta T)-\boldsymbol{u}(\boldsymbol{x},t)]-\frac{\mu\Delta}{\Delta T}\int_t^{t+\Delta T}\boldsymbol{u}+\frac{1}{\Delta T}\nabla\int_t^{t+\Delta T}p=\mathbf{f},$$

Because the the problem is periodic, $\boldsymbol{u}(\boldsymbol{x},t)=\boldsymbol{u}(\boldsymbol{x},t+\Delta T)$,

Then we can get, $-\mu\Delta\overline{\boldsymbol{u}^*}+\nabla\overline{p^*}=\mathbf{f}$,

In this equation, t can be a certain time point at any time period. ΔT stands for the period for any periodic behavior. $\overline{\boldsymbol{u}^*}$ and $\overline{p^*}$ stands for the time average solution for both velocity and pressure. $\boldsymbol{p}$ stands for the position vector and it is obvious to see that the velocity in the unsteady problem $\boldsymbol{u}$ is a vector that depends on both the position vector and time.

From the last equation in the equation 3.1, we can observe that the problem has been changed from solving the unsteady version of the equation stated in equation 2.1 to solving the steady equation stated in equation 2.2 with the velocity and pressure changed from time dependent to their time averaged version.

In this case of the stokes problems, the time-dependent problem can be surrogated directly with the steady problem to find the time-averaged solution with no errors.

This system is used later as the way to confirm the correctness of the algorithm of the solver.

3.2 The nonlinear case (Navier-Stokes)

The surrogate from the unsteady problem to the steady problem cannot be perfectly applied to the Navier-Stokes equations. Here is the deducton:

$$\frac{1}{\Delta T}\int_t^{t+\Delta T}(\rho\frac{\partial \boldsymbol{u}}{\partial t}+\rho\left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u}-\mu\Delta\boldsymbol{u}+\nabla p=\mathbf{f}),$$

$$\frac{1}{\Delta T}\int_t^{t+\Delta T}\rho\frac{\partial \boldsymbol{u}}{\partial t}+\frac{1}{\Delta T}\int_t^{t+\Delta T}\rho\left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u}-\frac{1}{\Delta T}\int_t^{t+\Delta T}\mu\Delta\boldsymbol{u}+\frac{1}{\Delta T}\int_t^{t+\Delta T}\nabla p=\frac{1}{\Delta T}\int_t^{t+\Delta T}\mathbf{f},$$

$$\frac{\rho}{\Delta T}[\boldsymbol{u}(\boldsymbol{p},t+\Delta T)-\boldsymbol{u}(\boldsymbol{p},t)]+\frac{\rho}{\Delta T}\int_t^{t+\Delta T}\left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u}-\frac{\mu\Delta}{\Delta T}\int_t^{t+\Delta T}\boldsymbol{u}+\frac{1}{\Delta T}\nabla\int_t^{t+\Delta T}p=\mathbf{f},$$

Because the the problem is periodic, $\boldsymbol{u}(\boldsymbol{p},t)=\boldsymbol{u}(\boldsymbol{p},t+\Delta T)$,

Then we can get, $\frac{\rho}{\Delta T}\int_t^{t+\Delta T}\left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u}-\mu\Delta\overline{\boldsymbol{u}^*}+\nabla\overline{p^*}=\mathbf{f}$,

(3.2)

The variables in this equation 3.2 has the same meaning as the previous deduction. However, the results is different since we can not factorized the gradient out of the integral of the term $\frac{\rho}{\Delta T}\int_t^{t+\Delta T}\left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u}$ because of the non-linearity. This shows that the time dependent Navier-Stokes equation (equation 2.3) cannot be perfectly surrogated by its steady version of the equation

(equation 2.4). At the same time, it also raises a problem: how well the problem can be surrogated, which can also be stated as how close is the term $\frac{\rho}{\Delta T}\int_t^{t+\Delta T}(\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u}$ and the term $\left(\overline{\boldsymbol{u}^*}\cdot\nabla\right)\overline{\boldsymbol{u}^*}$. In other terms: what is the error when we replace the time average of the nonlinear terms with the nonlinear term involving the time average? We will answer in the next Chapters.

Chapter 4

The numerical code

4.1 The FreeFem++ environment

The FreeFem++ is a popular partial differential equation solver for non-linear multi-physics system in 2D and 3D. Navier-Stokes equation, as a important physics problem, has its solver already been pre-built in the software. It also offers a large finite element lists. It also includes a first interpolation algorithms and a language for the manipulation of data on multiple meshes [4].

It is written in C++ and its language is C++ idiom [4].

4.2 Numerical test 1: flow in a cylinder

4.2.1 The meshes and boundary conditions

A figure of the meshes of the cylinder is shown in figure 4.1 below.

Figure 4.1: This is an example of the fine mesh that is going to be used in this experiment, the dotted lines describes the boundary of those tetrahedrons and they are where the solutions are derived.

In this research, the cylinder is assumed to be a pipe that is injected fluid from one side, that is, all the velocity of the fluid at the side boundaries for the cylinder should set to be zero because they are considered as the boundary of the pipes. The fluid is injected from one side and that is characterized by the injection condition on the z axis. The injection condition for the steady problem in the cylinder is set to be uIn, which is a variable that control the boundary condition. The boundary condition of the unsteady problem with the periodic boundary condition is set to be $uIn(1 + sin(2\pi t))$.

4.2.2 The algorithm to solve the steady problem

Since, in this case, steady problem is a non-linear problem, so some iterative converging method to find the root is needed to be applied here besides solving it on the mesh. Two methods, Picard and Newton, are used in this solver and they are explain below in the case of finding the root for a 1D problem and after that we'll explain why we use two methods together to reach to our roots.

Assume there is non-linear problem $x^2 = a$ and we are using the Picard iterative method to approach solution for x. We change the equation into the form below:

$$x^n = \frac{a}{x^{n-1}} \tag{4.1}$$

In this equation, x^n stands for the value of x at the nth iteration and x^{n-1} stands for the x

value at the (n-1)th iteration. Solving this equation iteratively and stop at a point where $x^n - x^{n-1}$ reaches to a small value, that is, when they are about to equal to each other, will lead us to the solution of x for the non-linear problem. This method's advantage is that it is generous about the initial guesses and even for a guess that is far away from the solution it will still converge in the end. However, it takes too many iterations to reach to the solution, especially at the positions that is really close to the solution.

In order to reach to the solution with less iterations, we need to have a method that will converge faster to the solution and this method, in our research, is the Newton way of finding the root. A sample graph of this solution is shown by figure 4.2 below :

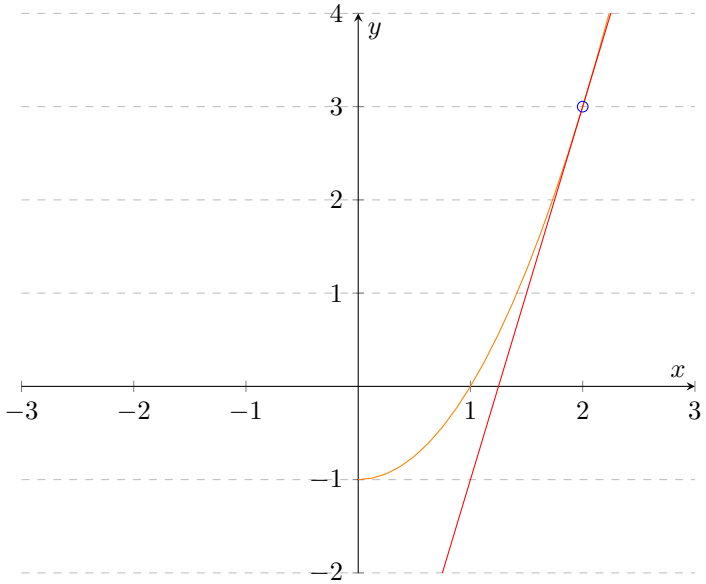


Figure 4.2: This is a graph showing how Newton find method in a non-linear problem. In this graph, the red curve is the tangent line of the orange curve at the blue point.

This figure 4.2 is illustrating how Newton method approaches its solution. We first have a initial guess and we find the point corresponding to that initial guess on the function of x in the space. After that, we take the tangent to that function at that point and find its intersection with the x-axis. After find that new x value, we find the point corresponding to that x value again on the function and redo the iteration again. This is the iteration for the Newton method is and it is proved to converge faster than the Picard method. However, it needs a strict initial guess for method to converge in the end for the solution. It requires the initial guess on the curve to be at the position of the curve that has the same concavity as the root.

Because both of these methods have fallback, we take a combination of those two methods to a better designed method to reach the root of the problem. We use the Picard method first to find a acceptable guess that is close to the root for the Newton method. Then, we used the Newton method to reach to an actual solution to the problem. Since if Newton method can converge faster with a good initial guess, this method can take less iterations to reach to the root.

This is prove by our test for those two methods in the code. When we use only the Picard method, it will take around 15 iterations to converge to the solution. When we use the Newton method, sometimes the iteration will not converge because of the bad guesses. At the same time, the combination of those two method will finally reach to the root with only 10 iterations.

4.2.3 The algorithm to solve the unsteady problem

To solve the unsteady problem, besides using the space discretization approximation, we also need to solve time step by time step to do the simulation. After having the solution on every time points, we take the time average of them to find the time average solution over one period. Since the boundary condition is set to be $uIn(1 + sin(2\pi t))$, the time period for one cycle is 1 second. The time steps is set to be identically separated in one period to be $1/Ns$, where Ns stands for the number of the time steps that is needed to be solved at. The result of the test is showing in the section 5.1.1.

4.2.4 Test the code's correctness in cylinder

As we mentioned in the section 3.1, the Stokes equation can be used as a way for testing the correctness of the solver of the problem. Before doing the test, we should first removed the nonlinear term from the problem defined and analyze the solution given by those two problems. We expect to see the time-averaged solution for the unsteady problem to be the same as the solution given from solving the steady problem in this test.

4.2.5 Variables set in the code for testing

There are three variables that is set in the code, represented by the global variable uIn, Ns and nn respectively. uIn will affect the amplitude of the boundary condition. Ns will affect the number of time steps that the unsteady will be solved on. nn will affect the mesh quality that is generated by FreeFem++ and this will affect the points of the solution to be solved on.

4.3 Numerical test 2: flow in a room

4.3.1 The meshes and boundary conditions

The mesh generated for the room is shown in figure 4.3 below:

Figure 4.3: This is an example of the fine mesh that is going to be used in this experiment for the model of the room, the purple lines in the graph stands for the boundaries of the tetrahedrons generated in this mesh.

In this figure 4.3, there are four parts in black. They represents three windows and one door in the geometry. The black rectangle in the back of this figure shows the position of the door and the three black rectangles in the front shows the positions of the windows.

In this experiment, fluid is set to be injected from the door and will exit from those windows.

4.3.2 The algorithm to solve the problem

If the algorithm for the solver for the cylinder works, it will also work for a different geometry. This means we can use the exactly same solver for the room and set the same variables in there. This time, the variable is represented by three different global variable. They are inflow, Ns and n. They represents the amplitude of boundary condition, number of time steps and the mesh quality respectively.

Chapter 5

Numerical results

5.1 Test case 1: Flow in a cylinder

5.1.1 The linear case: code testing

In order to test the code is correct, as we said in section 3.1, we need to test the code first to make sure the code is giving us the reliable result of the problem. In order to do this, we remove all the no linear part in the problem and run the code of the solver. It is better to do this experiment for the cylinder since the results on a regular shape is predictable and will not be affected by the irregular behavior of the meshes. Once it is confirmed to be correct on the cylinder geometry, it works on the other geometry because the solver doesn't change between geometries, the only thing changing between different shapes is the meshes and nodes.

After testing the code for the cylinder without those linear terms, it is found that the difference between the solution of steady problem and the time-averaged solution of the unsteady problem is less than a order of 10^{-13}. It is about the order of machine epsilon, which is around 10^{-16}, which means the algorithm of the solver is tested to be good to solve the problem with Stokes problems considering the size of the problem. By adding the non-linear terms to the solver, we can now sure that this solver will works for the full Navier-Stokes equation.

5.1.2 The nonlinear case

In these problems, as we described in section 4.2.5, we have several variables that can be tested on: mesh quality, the amplitude of the boundary condition, and the number of time steps while solving the full unsteady Navier-Stokes equation (equation 2.3). Changing those variables will lead us to different results. In the experiment, the variable that is not expected to change will always be controlled to be the same.

Changing the quality of the meshes

The variable nn in our code dictates the number of the points taken on the 3D model, which will then affect the quality of the meshes since the number of nodes and the number of the tetrahedrons are changing. The theory of the finite element method ensures that for $nn \to \infty$ with some regularity assumptions the error vanishes (in exact arithmetic). This effect is shown by the graphs below,

from which we can see the difference between the condition of nn = 3 and nn = 8. The amplitude of the boundary condition is set to be 1.0 as constant and the number of time steps is set to be 100, which means the time difference in this 1s period for every steps is set to be 0.01s as constant. After testing the data at $nn = 3, 4, 5, 6, 7, 8$ respectively , we have the data.

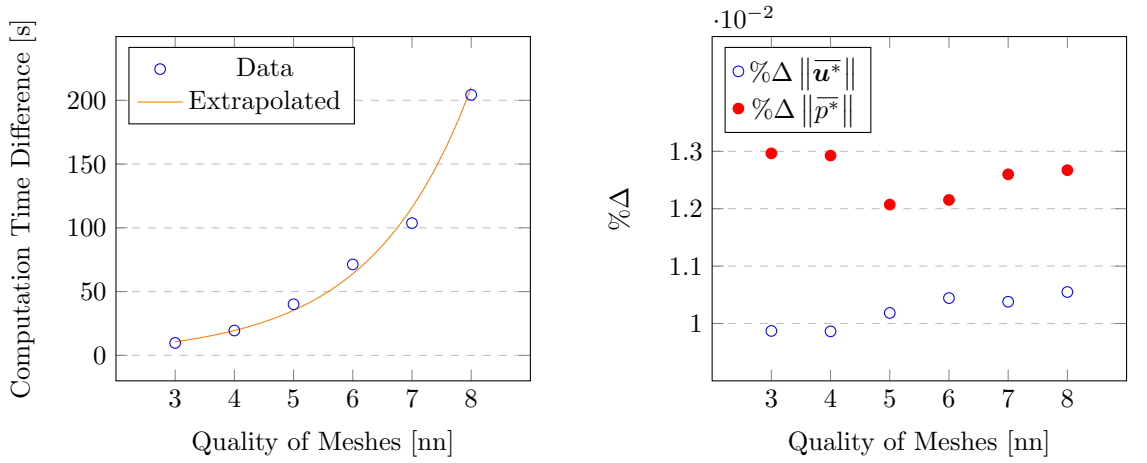
Since our goal is to find the effect of the non-linear term in this surrogate from the time dependent problem to the time independent problem, the important data in this table is the percentage difference of the spacial average norms and the time taken for the CPU to solving with those two different approaches . After taking those data out of the table, we have the table 5.1:

Table 5.1: The table shows the useful data in the test

nn(quality of meshes)	cpu time difference	$\%\Delta \left\|\overline{\boldsymbol{u}^*}\right\|$	$\%\Delta \left\|\overline{p^*}\right\|$
3	9.73	0.987%	1.30%
4	19.465	0.986%	1.29%
5	40.064	1.018%	1.21%
6	71.284	1.04%	1.22%
7	103.717	1.04%	1.26%
8	204.272	1.05%	1.27%

When observing the data, we can see that the time difference of the solving two problems increase significantly when the mesh qualities improve. It matches our expectations that because the size of the matrices to solve will increase significantly as the increasing in the number of nodes (solutions are taken at nodes and at the mid points of nodes), solving with time steps will cause an explosion in the cost of the algorithm. However, it seems that the fineness of mesh will not affect the percentage difference between the solution of the steady problem and the time-averaged solution of the unsteady problem as the percentage difference of both average norm of velocity $\left\|\overline{\boldsymbol{u}^*}\right\|$ and the average norm of pressure $\left\|\overline{p^*}\right\|$ will oscillating at the level 0.010 and 0.012 respectively.

To understand the relationship between data more clearly, we have plotted the graphs below according to the table 5.1.



(a) Computation Time Difference between Two Approaches vs Mesh Quality. The Extrapolated Equation of the Data Points is $1.8025e^{0.5948x}$.

(b) Percentage Difference between Two Approaches vs Mesh Quality Of Velocity $\boldsymbol{u}$ and Pressure p

Figure 5.1: The plotted graph for the data in table 5.1

From the figures 5.1a and 5.1b, we can see that only the differences in the time cost of CPU is increasing rapidly, in a rate that can be extrapolated by the exponential function, which is definitely a rate of explosive. It meets our expectation to see that it is increasing significantly and it is the reason that we're doing this surrogation to reduce the cost of computation and this seems to be meaningful. However, while the quality of the mesh is increasing, it seems that the percentage difference between the solution of the steady problem and the time-averaged solution of the unsteady problem are relatively stable (with the percentage difference of velocity oscillating around 1.25% and the percentage difference of pressure oscillating around 1.02%), which can be said that the fineness of the mesh will not affect how they're different to each other.

These differences, on the other hand, are due to the approximate computation of the average, the time-discretization and the rounding errors.

Changing the amplitude of the boundary conditions

In this numerical experiment we change the peak value of boundary condition in one period, which can also be said as the amplitude of the boundary condition. While changing the amplitude of the boundary conditions represented by Q in the code, we're keeping the other two variables to be the same to controlled the other variables. The mesh quality, represented by nn in the code, is set to be constant equal to 6 for this experiment. At the same time, the number of time steps is set to be 100, which means 0.01s interval, the same as the time steps number for the previous test.

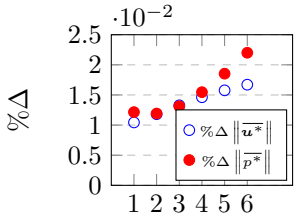
Since the qualities of the meshes is the same this time, we can take the time from the important data series since under this condition there is no difference in the number of equation solved. However, since the percentage differences, which is a core tested terms in this experiment, will also depend on the time-averaged solution of the unsteady problem which will indicate the absolute differences together with the percentage differences and is changing when the boundary condition is changing.

Here we have the tables for the necessary data about the percentage differences between the solutions and the amplitude for the boundary conditions.

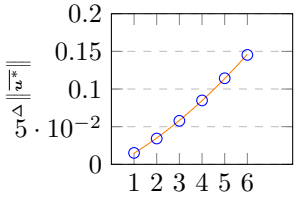
Table 5.2: These are the data obtained while doing the numerical experiment for changing the amplitude of the boundary conditions.

Q	$\left\|\overline{\boldsymbol{u}^*}\right\|$	$\left\|\overline{p^*}\right\|$	$\%\Delta\left\|\overline{\boldsymbol{u}^*}\right\|$	$\%\Delta\overline{p^*}$
1.0	1.45239	47.7902	1.04%	1.22%
2.0	2.90312	96.14	1.18%	1.19%
3.0	4.35265	145.034	1.33%	1.31%
4.0	5.80137	194.473	1.46%	1.55%
5.0	7.24957	244.473	1.58%	1.85%
6.0	8.69746	295.054	1.67%	2.20%

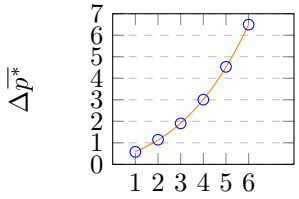
The results show a weak dependence of the errors on the amplitude of the boundary conditions. As a matter of fact, we argue that the change is basically due to a larger impact pf the time approximation and the rounding errors. It is worth noting that FreeFem++ has a bizarre management of the boundary conditions, that induces a large condition number of the matrices. We may speculate that also this factor my explain the weak dependence on Q. However, an error of 2% can still be considered acceptable, showing that the time average and the solution of the Stokes equations can commute. For a better understanding of the results, we show some graphs hereafter.



(a) The relationship of the factor for the amplitude of boundary condition (BC) with the percentage differences of $\left\|\overline{\boldsymbol{u}^*}\right\|$ and $\overline{p^*}$



(b) The relationship between the absolute difference of $\Delta\left\|\overline{\boldsymbol{u}^*}\right\|$ and the factor of the amplitude of the boundary condition. The absolute difference comes from the product of the second and fourth column in table 5.2. The extrapolated function is $0.0015x^2 + 0.0158x - 0.0026$.



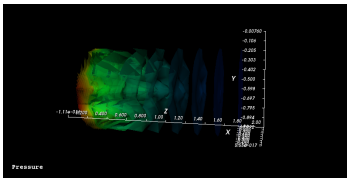
(c) The relationship between the absolute difference of $\Delta\left\|\overline{\boldsymbol{u}^*}\right\|$ and the factor of the amplitude of the boundary condition. The absolute difference comes from the product of the third and fifth column in table 5.2. The extrapolated equation is $0.0134x^3 + 0.0387x^2 + 0.3354x + 0.1982$ and the data perfectly fit with the cubic equation (with $R^2 = 1$)

Figure 5.2: The plotted graph for the data in table 5.2

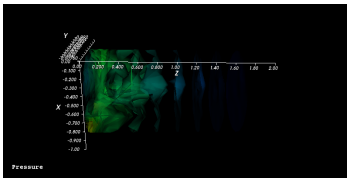
We can almost observe a linear relationship for the percentage error of pressure but since the solution of pressure is changing significantly, it is less meaningful to observe the relationship of

the percentage error for pressure and velocity(which even decrease in percentage error first). However, the results for the relationship between the factor of amplitude and the absolute difference in the solutions of velocity is far better than the expectation. They almost fit in the cubic and quadratic extrapolation perfectly. This suggests a strong relationship between the boundary condition's amplitude and the absolute difference between the solutions which means when constructing the equation that illustrates the surrogate of the Navier-Stokes equation with its steady version, the boundary condition's amplitude must be an important factor in that equation.

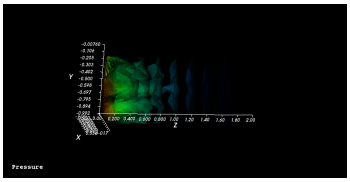
At the same time, the result can also be observed by the vector solutions manually on the model. Here we have the graphs for the iso-value planes for pressure in the cylinder and the point-wise vector solutions for those different boundary conditions



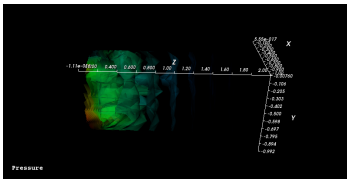
(a) Iso-value planes for pressure at Q = 1.0



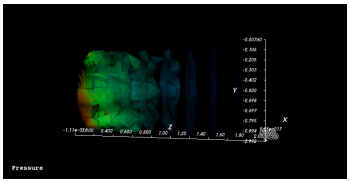
(b) Iso-value planes for pressure at Q = 2.0



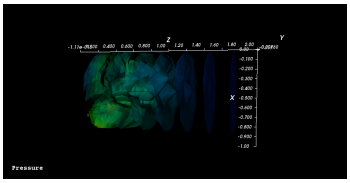
(c) Iso-value planes for pressure at Q = 3.0



(d) Iso-value planes for pressure at Q = 4.0

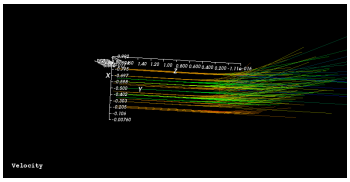


(e) Iso-value planes for pressure at Q = 5.0

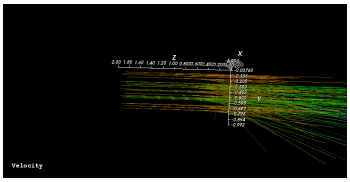


(f) Iso-value planes for pressure at Q = 6.0

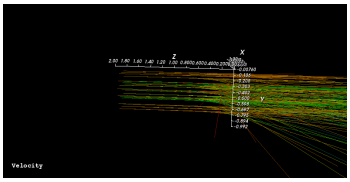
Figure 5.3: The pressure plots on the geometry



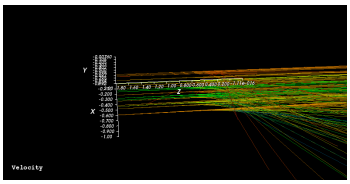
(a) The velocity point-wise vector plot at Q = 1.0



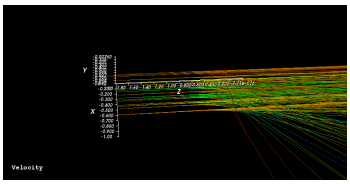
(b) The velocity point-wise vector plot at Q = 2.0



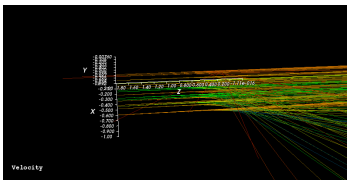
(c) The velocity point-wise vector plot at Q = 3.0



(d) The velocity point-wise vector plot at Q = 4.0



(e) The velocity point-wise vector plot at Q = 5.0



(f) The velocity point-wise vector plot at Q = 6.0

Figure 5.4: The velocity plots on the geometry

The results show that while the solution is expected for the most part, there are some unexpected boundary effects (the vectors deflecting at the boundary) probably due to the inaccurate management of the boundary conditions.

The effect of number of time steps on the differences

Now we cast the tests on how the number of the time steps will affect the differences of the solutions from two approaches. In this experiment, we will keep the mesh quality to be fixed at 6 and the factor of the amplitude of the boundary condition Q to be fixed at 1.0 to controlled the other variables. Since the time steps will affect the number of the matrices solved in the solver and the number of loops needed for solving the unsteady problem, the computational time here again becomes a relevant data for this numerical experiment. Since we need to make sure that the calculation stops at 1 period, which means the calculation will stop at 1s exactly to make the time-averaged solution of the unsteady problem comparable with the solution obtained by solving steady version of the problem (that is, to make sure the time-averaged solution to covered the whole period), we need to make sure that the time interval can be divides 1 to a integer to make sure it will finally reach to 1s at the final step. In this case, we take time intervals to be 0.02s, 0.04s, 0.10s, 0.20s

After doing the test, we have the data below as our data for the experiment:

Table 5.3: The data of the numerical testing on different time steps, Q is set to be 1 and mesh quality is set to be 6 for this experiment

time interval[s]	cpu time difference	$\%\Delta \left\|\overline{\boldsymbol{u}^*}\right\|$	$\%\Delta \left\|\overline{p^*}\right\|$
0.02	39.419	0.33%	0.33%
0.04	16.337	0.32%	0.46%
0.1	4.137	0.28%	1.05%
0.2	-0.735	0.18%	1.81%

From table 5.3, we can observe several patterns. The first main pattern that we can observe is that the time difference between the unsteady solver and the steady solver is becoming less and less since the steps needed for solving unsteady problems has be reduced because of the increasing in the time interval but the period does not change. There is even a negative time difference here in the table because as we explained in the section 4.2.5, In order to make the non-linear problem converge and find the solution, we use Picard and Newton's way of converge for the steady case, which cause several loop and in this case it is even takes longer than the calculating the 5-step long problem.

However, for the other terms in the table, it is obvious that $\%\Delta \left\|\overline{\boldsymbol{u}^*}\right\|$ is decreasing while the $\%\Delta \left\|\overline{p^*}\right\|$ is increasing. With the fact that the solution is not changing in this case, this will also be affect in terms of absolute. Since the time steps decreases, the time-averaged solution will be only reflected on less times points which will finally be calculated into the time-averaged solution. So it is hard to say whether the solution of $\left\|\overline{\boldsymbol{u}^*}\right\|$ and $\left\|\overline{p^*}\right\|$ will be more accurate or not since it is unpredictable which points and what will be the behavior of the solution at those points. The results show that: (1) with an appropriate selection for the time step, the errors between time average and steady solution is, in fact, negligible; (2) finer is the time step, much larger is the computational time between the two options, justifying the use of surrogate steady models to buy computational efficiency.

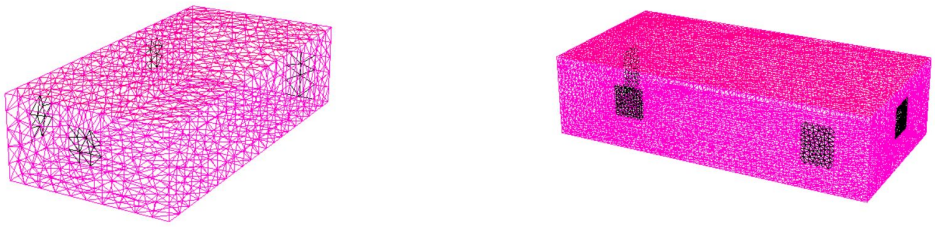
5.2 Test case 2: Flow in a room (Navier-Stokes Problem)

Using the geometry of a room is for finding how the geometry of the shape will affect the result of the simulation and how it will change the difference between two solvers. Since even we take the same nn for the mesh quality, we will get different numbers of tetrahedrons and nodes on the shape so it is hard to control all the variables between the geometries since the original accuracy . The only method reliable is to find how the rate of change of differences changes with the changing of those variables compared with those changes happened under the shape of a cylinder. It means that the main purpose of using this shape is to have cross comparison with the cylinder's case.

We will only test the dependence on the mesh quality for the sake of brevity.

5.2.1 Changing the quality of the meshes

Here we keep inflow to be at 100 and the number of time steps to be 10 (since the computational cost is too high under this irregular shape). We test the case under n = 1 and n = 2. The quality of the meshes is shown by the figure 5.5a and 5.5b.



(a) Mesh of the room when n = 1 (b) mesh of the room when n = 2

Figure 5.5: The different quality of the meshes of the roomn

The data tested is shown in the table 5.4,

Table 5.4: The table for the tested data of changing the quality of meshes for the room geometry

n	$\left\|\overline{\boldsymbol{u}^*}\right\|$	$\left\|\overline{p^*}\right\|$	$\%\Delta\left\|\overline{\boldsymbol{u}^*}\right\|$	$\%\Delta\overline{p^*}$
1	377.783	27949	19%	23%
2	733.103	49479.8	17%	24%

From the table, it can be easily observed that the solutions for the pressure becomes really large. Since in this case, even the solution itself is quite different when the quality of the meshes is different, we can conclude that the when the geometry becomes more complicated, the fineness of the mesh itself will have a large effect on the solution of the solvers. However, comparing to the change in the solutions(almost 1.8 times change in this case), the percentage difference changes are relatively tiny(with only 1% to 2% change) which means the quality of the meshes will not affect the $\%\Delta\left\|\overline{\boldsymbol{u}^*}\right\|$ and $\%\Delta\overline{p^*}$ a lot, which leads to a similar results for the solutions in cylinder.

5.2.2 Changing the inflow factor of the boundary condition

Now keeping the quality of meshes and number of time steps unchanged, we test how the inflow conditions will change the difference between two approaches. We take the inflow condition to be 100 and 50 respectively for the amplitude of the inflow boundary condition.

Here is the table for the data obtained from the experiment:

Table 5.5: The data obtained from the numerical experiment of testing the inflow on the percentage difference

inflow	$\left\|\overline{\boldsymbol{u}^*}\right\|$	$\left\|\overline{p^*}\right\|$	$\%\Delta\left\|\overline{\boldsymbol{u}^*}\right\|$	$\%\Delta\overline{p^*}$
50	346.419	14944.6	21.04%	20.95%
100	733.103	49479.8	17.21%	24.23%

From table above, we can clearly see that we cannot find a pattern that is even close to the percentage difference presented in the cylinder. We cannot see monotonicity in the change of the percentage difference between the two solutions for $\overline{p^*}$ and $\left\|\overline{\boldsymbol{u}^*}\right\|$. However, with the fact that the absolute difference in the cylinder has a really strong relationship with the amplitude, we can conclude that the target geometry will has a larger effect in the construction of the mapping from the solution of the steady version of the problem to the unsteady one.

5.3 Discussion

5.3.1 For the nonlinear term in the equation

In this experiment, we found that there is a good pattern for the differences between those two solutions found in the cylinder but not found in the geometry of a room when changing the boundary condition. This is because the cylinder has analytically a zero nonlinear term. The nonlinear term is shown by equation 5.3.1:

$$(\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_x\dfrac{\partial \boldsymbol{u}_x}{\partial x} + \boldsymbol{u}_y\dfrac{\partial \boldsymbol{u}_x}{\partial y} + \boldsymbol{u}_z\dfrac{\partial \boldsymbol{u}_x}{\partial z} \\ \boldsymbol{u}_x\dfrac{\partial \boldsymbol{u}_y}{\partial x} + \boldsymbol{u}_y\dfrac{\partial \boldsymbol{u}_y}{\partial y} + \boldsymbol{u}_z\dfrac{\partial \boldsymbol{u}_y}{\partial z} \\ \boldsymbol{u}_x\dfrac{\partial \boldsymbol{u}_z}{\partial x} + \boldsymbol{u}_y\dfrac{\partial \boldsymbol{u}_z}{\partial y} + \boldsymbol{u}_z\dfrac{\partial \boldsymbol{u}_z}{\partial z} \end{pmatrix} \tag{5.1}$$

In this equation, because the fluid velocity in the cylinder will only depends on x axis and y axis but the velocity $\boldsymbol{u}$ in those two directions will be zero all the time, which means $\boldsymbol{u}_x$, $\boldsymbol{u}_y$, $\dfrac{\partial \boldsymbol{u}_x}{\partial z}$, $\dfrac{\partial \boldsymbol{u}_y}{\partial z}$ and $\dfrac{\partial \boldsymbol{u}_z}{\partial z}$ will all becomes zero in this way. This will lead the non-linear term to become zero in the cylinder's geometry. Numerically, this term will approach to zero which means it will not have a significant effect on the solution. In that condition, it seems that the amplitude of the boundary condition will have a small polynomial effect on the differences between those two solutions. Since this behavior is not observed while changing the boundary condition of the room, we can say that the amplitude of the boundary condition will dominates the differences between those two solutions only at small non-linear terms.

5.3.2 For the irregular geometry

Comparing with the relative differences between those two solutions for the room and the differences for the cylinder, we can see that the relative differences between the solutions for the room is a lot larger. This means that when the geometry is irregular, the nonlinear behavior caused by the geometry will dominates the differences. In this condition, unless we're looking for a "quick-and-dirty" solution for the results, we should not used the steady problem as a surrogate for the unsteady problem or we can find a relationship between those two solutions with geometry and the amplitude of the boundary condition as factors.

5.3.3 Future directions

In order to test how the value of the non-linear term can affect the differences between the solutions, we can run this simulation on a cylinder-like geometry. Comparing the results of that with the results from the perfect cylinder can give us insight on how a slight change in the nonlinear term will affect the differences between solutions. We can also run the simulation directly on the geometry for an aorta to test whether this surrogate is a good approach to solve realistic problems in human vessels.

Conclusion

In this numerical experiment, we reach to a conclusion that in a irregular geometry, the relative differences between the solutions solved from steady Navier-Stokes equations and the time-averaged solutions from unsteady Navier-Stokes equations with periodic boundary conditions will be dominated by the geometry itself. In a regular geometry, when the non-linear term is small, the amplitude of the boundary condition will have a polynomial effect on the differences.

References

[1] P. J. Blanco et al. "Comparison of 1D and 3D Models for the Estimation of Fractional Flow Reserve". In: *Scientific Reports* 8.1 (2018), p. 17275. ISSN: 2045-2322. DOI: 10.1038/s41598-018-35344-0. URL: https://doi.org/10.1038/s41598-018-35344-0.

[2] Stanford Health Care. *Left Ventricular Assist Device (LVAD)*. [Online; accessed 19-March-2019]. URL: https://stanfordhealthcare.org/medical-treatments/l/lvad.html#about.

[3] K Gashi, EMH Bosboom, and FN van de Vosse. "The influence of model order reduction on the computed fractional flow reserve using parameterized coronary geometries". In: *Journal of biomechanics* 82 (2019), pp. 313–323.

[4] F. Hecht. "New development in FreeFem++". In: *J. Numer. Math.* 20.3-4 (2012), pp. 251–265. ISSN: 1570-2820.

[5] William L. Hosch. *Navier-Stokes equation*. [Online; accessed 28-March-2019]. URL: https://www.britannica.com/science/Navier-Stokes-equation.

[6] ibidi. *The Different Types of Flow*. [Online; accessed 31-March-2019]. URL: https://ibidi.com/content/286-the-different-types-of-flow.

[7] B.J. Kirby. *Micro- and Nanoscale Fluid Mechanics: Transport in Microfluidic Devices*. Cambridge University Press, 2010. ISBN: 9781139489836. URL: https://books.google.com/books?id=y7PB9f5zmU4C.

[8] M. Labrosse and M.R. Labrosse. *Cardiovascular Mechanics*. Taylor & Francis Group, 2018. ISBN: 9781138197237. URL: https://books.google.com/books?id=NwqbnQAACAAJ.

[9] Pavan Kalyan Nandamuri. *What is the difference between laminar and turbulent flow?* [Online; accessed 31-March-2019]. URL: https://www.quora.com/What-is-the-difference-between-laminar-and-turbulent-flow.