Possibility of developing CFD web application to optimize sedimentation tank in WWTP

Sifat Munim Tanin

Norwegian University of Life Sciences
Department of Mathematical Science and Technology
Master Thesis 60 credits 2014

Norwegian University of Life Sciences
MDCCCLIX

Possibility of developing CFD web application to optimize sedimentation tank in WWTP

Master Thesis

by

Sifat Munim Tanin

Norwegian University of Life Sciences
Ås, Norway
March, 2014

Acknowledgements

I am immensely grateful to my supervisor, Harsha Chandima Ratnaweera, who has always been forthcoming with his constant support & patience. I am equally thankful to be given the opportunity to finish the master's degree in NMBU.

I would also like to express my love & appreciation to my family for their hearty assistance & encouragement whenever it was needed.

Sifat Munim Tanin
March 2014
Ås, Norway

Abstract

Computational fluid dynamics (CFD) is used to describe temperature, velocity and different fluid properties of different structure by using various numerical methods. It gained a huge popularity in the last few decades. Inadequate knowledge about CFD has become a major barrier for its application in different sector. In this study, it was attempted to present CFD in an easier and more accessible way to WWTP workers. One web platform has been conceptualized to be developed in such a way that it can simulate fluid flow in a sedimentation tank. Different open source tools have been used to build this platform. Django web development framework based on python programming language has been used for running simulation in a web platform. Simple simulation has been run in the web framework to make it applicable for CFD simulation. To run a complex multiphase flow simulation for a sedimentation tank, some open source CFD applications were needed to be integrated in the web platform. Those applications will run in the server machine without any direct contract with the user. The user will only get a simple web interface to run the simulation in different types of sedimentation tank and be able to take decision about the design.

Contents Page no.

Chapter 1: Introduction 1
1.1. Motivation 1
1.2. Objective of the study 2
1.3. Methodology 3
1.4. Outline of the paper 4
Chapter 2: Background 5
2.1. CFD in web application 5
2.2. Application of CFD at sedimentation of WWTP 7
Chapter 3: Technical background 9
3.1. Sedimentation tank 9
3.2. Optimization of sedimentation tank 11
3.3. Web development for CFD 12
3.3.1. Web application and desktop application 12
3.3.2. Open source and Commercial application 13
3.3.3. Open source programming languages 14
3.3.4. Importance of web development framework 15
Chapter 4: Experimental design 16
4.1. Python as a programming language 16
4.2. Web development in python 16
4.3. Web development framework 17
4.3.1. Django 18
4.3.2. Setting up Django in Computer 18
4.3.3. MTV 20
4.3.4. Model 20
4.3.5. Template 21
4.3.6. View 22
4.3.7. Settings.py 22
4.3.8. URL pattern 22
4.4.Python Libraries for CFD 23
4.4.1. Numpy 23
4.4.2. Plotly 24
4.4.3. SciPy 24
4.4.4. Matplotlib 24
4.4.5. Setting up NumPy and Matplotlib in Django 25
4.5.CFD simulation using Neiver-Stokes equation (cavity flow) 25
4.5.1. Navier-stokes equation 25
4.5.2. The finite difference method 26
4.5.3. Finite difference method in Navier-stokes equation 29
4.5.4. Naiver-stokes equation in Django 30
4.5.4.1.URL pattern 30
4.5.4.2.User interface 30
4.5.4.3.Views.py 31
4.6. Developing web application for sedimentation tank 33
4.6.1. Different models for multiphase flow simulation 33

4.6.2. Simulation by desktop application 34
4.6.3. Web structure for CFD simulation 35
4.6.3.1.Back-end structure 36
4.6.3.2Front-end 40
Chapter 5: The way forward 42
Chapter 6: Result 44
6.1. Cavity flow by naiver-stokes equation 44
6.2. Simulation for sedimentation tank 46
6.3. Architecture of the web application for sedimentation tank 52
Chapter 7: Discussion 53
Chapter 8: Conclusion 56
References 58

List of figures Page no

Figure 1: Django web development framework structure 23

Figure 2: Cartesian grid 26

Figure 3: User interface to input data for cavity flow simulation 31

Figure 4: Processes in Django for simulation 32

Figure 5: Cavity flow in different time steps 45

Figure 6: Volume fraction of the sediment after introducing slope and baffle. 47

Figure 7: Volume fraction of the sediment from the bottom area. 47

Figure 8: Cross section of the volume fraction of sediment using 2 baffles. 48

Figure 9: Login page in the user interface 49

Figure 10: Main workflow page of the user 50

Figure 11: Data input page for new project 51

Figure 12: Structure of the web application for CFD applications 52

Abbreviations

***ε* -** Epsilon

2D – Two Dimension

3D – Three Dimension

CAD – Computer Aided Design

CFD – Computational Fluid Dynamics

CGI – Common Gateway Interface

CPU – Central Processing Unit

CSS – Cascading Style Sheet

DB - Database

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IDE – Integrated Development Environment

JPEG – Joint Photographic Experts Group

MTV – Model Template View

MVC – Model View Controller

PDE – Partial Differential Equation

PDF – Portable Document Format

PHP - PHP: Hypertext Preprocessor

PNG – Portable Network Graphics

RSM – Renolds Stress Model

SVG – Scalable Vector Graphics

UI – User Interface

USB – Universal Serial Bus

WSGI – Web Server Gateway Interface

WWTP – Wastewater treatment plant

WWW – World Wide Web

Chapter – 1
Introduction

1.1. Motivation

One statistical graph from World Bank shows that the world population has increased more than 1 billion between the years 2005 to 2012 (Population www… 2014). With this high rate of population growth, wastewater production is increasing simultaneously. Wastewater purification is now a major challenge for every community. Using sedimentation tank in wastewater treatment plant (WWTP) has practiced from long ago (Shahrokhi et al. 2013). So better solution to optimize hydraulic efficiency in sedimentation tank has become a challenge for different field of studies. Different technologies have been implemented to improve treatment process.

Computational fluid dynamics (CFD) has brought forth a new era in industrial design and research. At first this field of study revolved around the areas of high-technology engineering, but with the progression of time and age it has been quickly embraced as a go to method for solving complex problems in engineering sector. Originally it began with the studies of fluid mechanics and heat transfer. But now they are widely associated with other areas such as chemical, civil and environmental engineering (Tu et al. 2007). Due to steady advancement of computational simulation in CFD it is now possible to design better equipment, which in turn results in lesser operating cost. It has also made a considerable impact on environmental pollutant reduction.

With this successful achievement in CFD, it is also coming forward with its own ideas to increase the efficiency of sedimentation tank design. Running CFD simulation for different design is helping the engineer to construct a better possible solution for wastewater treatment. In the sedimentation tank CFD can suggest improved design for many practical problems. For example, in many sedimentation tanks dead zone occupies some area, which reduces the sedimentation zone. It causes the efficiency loss in the tank. By using CFD, these zones can be examined and better designs can be proposed. However, CFD experts who have clear knowledge about fluid mechanics and numerical calculation behind the computational process mainly do CFD simulations. It is not as available and understandable to the general people with general expertise. It is also difficult to become an expert in CFD for those are related to WWTP and works for the improvement in different section of WWTP. Therefore, it becomes necessary to bring the CFD for WWTP in an easy-to-access and understandable medium. On the other hand, setting up a CFD application in a desktop platform could be difficult for the WWTP experts.

Through the innovative development in information technology, web application and services have gained more popularity. To meet the requirement of different people in different sectors, different types of web applications are being developed. Web development for CFD with a better user interface (UI) can run the complicated simulation while keeping the complex numerical calculations behind the scene when it gives output. If any user-friendly platforms, like web browsers work as a client-side platform for running the CFD calculation, it solves the complexity of setting up CFD application in their computer. This could be understandable, easy to use and preferable to WWTP experts. So they can take decisions depending on the result without understanding the internal intricacy. This initiative can expose new possibilities for using complex CFD in design or optimizing the sedimentation tank at WWTP. Developing an open source web application for sedimentation tank design in wastewater treatment can help the WWTP expert more readily. They can access the application for free and can work for further development if needed. Web developers from different area can work with the code and can contribute their productivity easily within a structured framework. It would be easier than developing an application from scratch.

1.2. Objective of the study

The main purpose of this study is to make CFD user-friendly, easy to access and more practical to the WWTP specialists with limited competence in CFD and advanced modeling tools. With such a tool, they can evaluate the shortcomings of existing designs and evaluate improvements with simulating various physical changes. The ultimate goal is to find the way for developing a web structure using a popular programming language that can manage and run the numerical calculation of CFD simulation for sedimentation tank. The user can input the primary data to run the simulation in a user-friendly website. The website has the capacity to grab the data from the input form and put it in the appropriate place for logical calculation. Logical code in the website will analyze the data and generate a figure explaining the fluid flow and the pressure term.

The objective of the study can be stated as follows:

- Bridge the gap between the complexity of computational fluid dynamics and the WWTP workers by making CFD simulation accessible through web interface.
- Make a web interface for the sedimentation tank in a way that will hide the CFD complexity from the general user by taking basic inputs and doing most of the computation on the server.
- Making a computational architecture using different CFD tools, which will run simulation for the sedimentation tank on web server.

1.3. Methodology

The methodology of the study can be divided into few different steps.

1. Understand the fluid flow behavior in sedimentation tank and different physical problems during sedimentation process.
2. Choose a programming language
    - Which is popular to the developers for developing CFD application.
    - Which could run the CFD application and output the figure of fluid flow pattern understanding the numerical calculation. It could be chosen by reviewing previous works done by the developers.
    - By examining the necessary plug-ins needed for CFD simulation.
    - Which have better ground for web development than others.
3. Develop a web application using necessary libraries and plug-ins, which can run the simple CFD simulation and can output the figure depending on the result found.
4. Find a possible way for developing CFD web application, which can run simulation in different structure of the sedimentation tank. The user interface of the web application will be simple and understandable to the user.

1.4. Outline of the paper

The paper has been structured in the following way:

Introduction explains why this topic is necessary and the possible achievement. It also explains the objective of the study and how the work has to be done.

Chapter 2 contains the background that discusses previous works related to this study.

Chapter 3 discusses the technical knowledge about the study and the possible tools for developing web application

Chapter 4 discusses about the experiments done during the thesis works.

Chapter 5 discusses about the future works needs to be done to complete the web application and possibility of running CFD simulation for the sedimentation tank.

Chapter 6 discusses about the result have found from the thesis work.

Chapter 7 discusses about the critical analysis of the study.

Chapter 8 illustrates concluding remarks of the study.

Chapter – 2
Background

2.1. CFD in web application

CFD applications are like a virtual laboratory that has the capability to visualize fluid flow in a medium by running qualitative or quantitative calculations. It gives an in-depth flow pattern that is difficult to find out through a traditional experiment. It cannot replace the actual experiment completely because of assuming too many input data and accuracy of choosing proper mathematical model for simulation. But it can reduce the cost of the experiment significantly. Different developer and software companies have tried to develop CFD applications in web platform.

Altair has developed a cloud solution of a CFD application named CFDCalc by the help of AcuSolve solver. In this solution, the user can access CFD technology or can take advantage of the framework to build his application. Its target is to solve two types of CFD problem i.e. fluid mixing and thermal cooling. It does not require CFD expertise from the user and can help him save the money required for purchasing necessary software and hardware for simulation. (Altair announces www… 2013).

In CFDCalc, user need to specify the parameters in a form and the application uses AcuSolve solver to model it, mesh generation and run the simulation to solve the flow problem. It has developed three solution packages i.e. Heat sink calculator, SMX mixer calculator and simple pipe calculator (Calculators to www… 2013).

- Heat sink calculator uses the finite element method to improve the performance of convective and conductive heat transfer. It investigates the temperature and the heat flux for required places. It has the capability of CAD model generation, CFD mesh generation and the visualization of processed CFD data. (Heat Sink www… 2013).
- SMX mixer calculator can simulate the high velocity fluid from different types of inlet such as normal, j-type and t-type inlet configuration. It also uses Hagen-Poiseuille equation to measure the pressure drop in an empty tube (SMX mixer www… 2013).

- In the simple pipe calculator, the workflow shows five different steps. It uses python programming language to run the application. It uses different python script to generate the CAD model, mesh file, AcuSolve input file, images, reports and web interface. It uses one shell script to maintain the batch job like AcuConsole automation, AcuSolve and AcuReport. Python script called SimplePipe.py generates CAD file. AcuConsole helps for further modification of the CAD file and mesh input. AcuReport create the .pdf file as a report. (CFDCalc simple www… 2013).

OpenFOAM introduces an open source CFD software package with different features like chemical reaction, heat transfer, turbulence, and electromagnetics. It has its own meshing tools and a large amount of solvers for different engineering problems. This package now supports only linux operating system (Features of www… 2011-2014). OpenFOAM is developed by C++ and to modify or further development C++ knowledge required (Jasak et al. 2007).

Pythonflu is a OpenFOAM API that have the features of OpenFOAM and has developed in python programming language. It shows the same performance as OpenFOAM and allows developer to use pure python classes rather than C++ programming. Pythonflu have some solvers those are similar to OpenFOAM but completely written in python, which show the same OpenFOAM functionality. Another feature of the pythonflu is the SALOME functionality. SALOME is a mesh generation software that can generate the CAD and can output the meshed structure from the CAD file. Pythonflu helps to combine these two application and works as a bridge between them to run the simulation. It is also built for linux platform. (pythonflu www… 2010)

Dacolt has developed CFD webapp for combustion engine. These web apps are available for both desktop and mobile devices. (Dacolt web www… 2004-2014)

Another web-based system for computational fluid dynamics has been developed using OpenFOAM (Sempolinski et al. 2012). For mesh generation it uses python library and for front-end it uses php programming language.

Another cloud based CFD simulation software has been developed by engineering.com named Ciespace. To understand the simulation it also developed a workflow engine that explains the steps that occur behind the scene. It uses solvers from OpenFOAM. This application does not require any client site platform and can run in normal browser like firefox or crome. (New cloud-based www… 2013)

CFD application in web platform is not as popular as desktop CFD application. Web applications those are developed for CFD simulation normally run simple simulations. Python programming language is broadly used for developing this kind of application. In desktop application like OpenFOAM, C++ have used for development. But it has a python API for python developer.

2.2. Application of CFD in sedimentation of WWTP

Sedimentation is a very common process in the wastewater treatment plant and is being practiced for long time. The main mechanism of the sedimentation tank is to separate the suspended solid from the water by gravitation. It could depend on different factors like climatic condition, flow rate, the structure of the inlet and outlet, structure of the sedimentation tank, method used for the removal of sludge etc. (Kawamura 1991). After a specific time interval, clean water is taken out form the tank for further treatment. The performance of the sedimentation tank is very important for the next treatment process. So many researchers focus on the efficiency of the sedimentation tank. To measure the performance and to improve the design of the sedimentation tank, CFD has practiced with other quality improvement methods. Particles carried by the wastewater into the tank for sedimentation can influence the performance of the tank. A simulation was run considering different particle diameter and volume fraction to understand the interaction between primary and secondary phase and its impact on the efficiency of the sedimentation tank (Tarpagkou & Pantokratoras 2013).

Introducing baffle in the sedimentation tank can increase efficiency. A study was done by setting up different number of baffle in the sedimentation tank and studying the consequences (Shahrokhi et al. 2012). Another simulation was done simulation by changing the location of the baffle and studying the efficiency of the sedimentation tank (Shahrokhi et al. 2013). Further advanced work has been done by studying the performance of the sedimentation tank through using transverse and longitudinal baffle (Wills & Davis 1962).

Temperature variation between the influent and the tank content can influence the direction of the circular current and as a result show impact to the suspended solid concentration (Goula et al. 2008). Studying of Geometries for sedimentation tank and application of CFD in different structure can suggest a cost-effective and better design solution (Stamou 2008).

Different models were implemented for sedimentation tank. The k-ε turbulence model equation was implemented to understand the flow pattern of the sedimentation tank (Schamber & Larock 1981). Particle fluid interactions were studied in different perspective (Righetti & Romano 2004). In the primary sedimentation tank the concentration of solid is limited. So one study was done by using a fixed settling and particle velocity (Imam et al. 1983). Momentum and solid concentration equation has been solved for the sedimentation tank by using 2D model (Stamou et al. 1989). For rectangular sedimentation tank, simulation was done for flow field and suspended solid concentration (Wang et al. 2008).

Changing different factors those are related to suspended solid formation could do optimization of the sedimentation tank. Another effective way is to use baffle in the proper place. Structure of the sedimentation tank also could be changed for getting more efficiency.

Chapter - 3
Technical background

3.1. Sedimentation tank

In a wastewater treatment plant, sedimentation tank works as a unit. In most cases sedimentation tank is the first treatment process. Sometimes in an industrial wastewater treatment plant, sedimentation tank is used as a second step. In this case screening process works as the first step to remove the large particles.

Some wastewater treatment plants have two types of sedimentation tank depending on their functionality, primary and secondary. Primary sedimentation tank is used to remove grit. On the other hand, secondary sedimentation tank remove flocs, which is produced by chemical coagulation process (Rodríguez López et al. 2008). To ensure water quality, sedimentation tank has a vital role in the treatment plant. Flow pattern in a tank influences the performance of the sedimentation process and to understand it & achieve better design, studies have been done over real tank in some studies (Rodríguez López et al. 2008). On the other hand, in a secondary sedimentation tank, different transport mechanisms work simultaneously, which makes it difficult to understand.

Depending on the shape, sedimentation tank can be divided into two types namely circular sedimentation tank and rectangular sedimentation tank. In a rectangular sedimentation tank normally inlet have connected in the one end and outlet have installed at the other end. When water reaches to the outlet, large amounts of particle have sediment by gravitational force. However, in a circular sedimentation tank inlet have installed at the bottom of the tank and the outlet have installed at the surface of the tank.

A sedimentation tank can be divided into four different zones i.e. inlet zone, settling zone, sludge zone and the outlet zone.

Inlet zone

Water have distributed with a controlled velocity from the inlet zone. It prevents turbulence flow. Water should be distributed evenly from the inlet zone. Improper distribution could cause short-circuiting. To ensure proper distribution of water inlet could be found in different structure. One of it is stilling wall, which is also known as perforated baffle wall.

Settling zone

Water enters to the settling zone through the inlet zone. Here the velocity of the water needs to low and the even distribution of the water needs to be ensured. This environment helps most of the folks settled down. It controls the water flowing out of the sedimentation tank.

Outlet zone

It controls how the water have discharged and distributed from the sedimentation tank. It also has a significant role to prevent short-circuiting. It also controls the water level of the sedimentation basin. Normally in a sedimentation tank baffle could be found in the in the outlet zone. Installing baffle prevents clogging and the floating materials entering into the next treatment process.

Sludge zone

Sludge zone is found in the bottom of the sedimentation tank where velocity is very becomes very low to keep the suspension of the solid. A drain is available in the bottom of the basin to remove the sludge from the sedimentation tank. Sludge has removed by automated equipment in many WWTP. In some cases sludge has removed manually. If WWTP follows manual sludge removal process, sedimentation tank needs to be cleaned twice a year.

Pollutants that arrive in a treatment plant mixed with water can vary depending on different situations. They could differ depending on the surroundings (where wastewater generate), climatic condition, population etc. When the snow melts or rainfall occurs, large variety of pollutants (led, zinc, copper, cadmium, chromium, nickel, nutrients, organic compound, de-icing agent) are introduced in the sewer system, which finally reach at the treatment plant (Hvitved-Jacobson & Yousef 1991). Fluid flow in a sedimentation swells up and become more complicated due to such water intrusion. Furthermore, continuous rising of water consumption rate increases pressure on the treatment plant to treat more and faster. To defeat this situation, wastewater treatment plant is growing larger from time to time. Optimization of different treatment unit in the plant has also become a major concern.

3.2. Optimization of the sedimentation tank

Different physical problem can occur during the sedimentation due to its structure and the flow pattern. Some of the significant problems have been discussed below that are needed to be considered for optimization.

Plug flow

Plug flow occurs when the velocity of the fluid remains constant in any part of the flow zone. An ideal plug flow is not possible in sedimentation tank. Flow pattern in an inlet area have changed when it comes into to the sedimentation tank (Maus & Uhl 2010). In the output are the flow have changed also. The deviation form the ideal plug flow in the sedimentation tank can explain the efficiency of the sedimentation tank (Maus & Uhl 2010). High deviation causes low efficiency in the sedimentation tank. Because the flow pattern of the sedimentation tank is as close as to the plug flow, particle sedimentation process occur more efficiently.

Coagulation/ flocculation process

Efficiency of a secondary sedimentation tank depends on the coagulation and the flocculation process. Flocs settling in the sedimentation tank mostly depend on their size, shape and density. Flocs those have low density due to irregular shape cause problems in the settlement. Another significant difficulty could occur in the inlet and the outlet. If the inlet of the tank has sharp bends, it could break the flocs into smaller particles. Flocs could also become small in size if they are discharged from a high level to the water basin. To increase the efficiency of a sedimentation tank, optimum settling should be done in the sedimentation tank.

Short-circuiting

Short-circuiting occurs in a sedimentation tank when part of the fluids flow higher or lower than a normal flow. In this case a portion of flow reaches the outlet quicker than the other flow part. When it happens, flocs do not get enough time to be settled. Short-circuiting happens mainly because of the geometry of the sedimentation tank. It also happens due to the position of the inlet and outlet. Improper distribution of flow also causes short-circuiting. Proper design of the sedimentation tank with an accurate position of inlet and outlet could solve most of these problems.

Dead zone

Dead zone in sedimentation tank is an area where circulation occurs. It reduces the area for particle sedimentation. Efficiency of the sedimentation tank decreases with the for having dead zone in different paper. It mainly found in the corner of the sedimentation tank. To reduce dead zone different techniques have developed. Using baffle in the sedimentation tank is the most common technique in the to reduce dead zone.

Retention time

Retention time in a sedimentation tank need to be considered for optimization. Particle should have to get proper time to be settled. By examining the sedimentation tank and the fluid quality, proper retention time need counted. This application can give a far better result from the sedimentation tank.

Temperature

Temperature also influences the efficiency of the sedimentation tank. With the reduction of the temperature, particle settlement becomes slower. So the retention time need to increase when the temperature becomes low. Doses of the coagulant also need to be changed with temperature. On the other hand, when the temperature reduces flow also reduces and the flocs get enough time to settled.

3.3. Web development for CFD

3.3.1. Web application and desktop application

Computer software or application is a part of computer system, which helps the users to solve their everyday problem. It can be defined as a package of instructions for the CPU to follow and help the CPU understand our needs. So it works as a bridge between user and the computer.

Instructions provided from us to the computer can be divided into two parts. One is operating system and the other is software application or program. Operating system is a bundle of primary instructions those are approximately same in every computer.

Software or application personalizes the computer for a specific user. It could be different depending on their types, categories, compatibility etc. Some are developed for maintaining the hardware and on the other hand others are developed to fulfilling user's specific needs.

If we think about our everyday used software, there are two categories of software seen. One category of application runs in the desktop and is called desktop application. Another categories of application run from the web server are web application. Web applications use their codes or other types of data from server. They might have one platform in the desktop but they access its major logical instruction from the server. Now-a-days web applications are becoming more useful and available to everyone because of its easy accessibility. Desktop applications get all its instructions to analyze data from the personal computer. Now-a-days desktop applications are being developed to solve more complex problems those are not easy to run in the web platform.

3.3.2 Open source and Commercial application

The key feature of the open source application is that it provides source code with application. User or developer has opportunity to change the code depending on his or her interest (Raymond 1999). With this kind of licenses, software have developed in different segments such as operating system, system software, and end user products for example software for writing documents or music software etc. Open source applications are becoming popular more and more each day. For example, if we consider web server, Apache achieved 60% market share in 2005 (von Krogh & Spaeth 2007). As the user get the source code, he could change the code for new hardware and could make the application compatible for different platform. So codes does not loose its usability in the long run. Due to the right to distribute, modify and develop, this application can be shared in the large community. Different people can add new functionality to the application and can share with others. So the application gets its development in time without any effort of the original developer. As it is free, large population of the user make a big market for the product, which attracts other developers to work with it for further improvement.

On the other hand, if it is a commercial application and if the program vendor decides not to develop the application for the future platform, the application becomes useless to the user and the user needs to find another software for new platform. Consider a large financial organization. They could have different financial software to maintain the workflow. If the software company, which is responsible for the application, but stops working, other software company have no right to access the application. For this reason, different big organizations are becoming interested in open source platform.

When a software company develops commercial software they face the pressure of competition with other similar products. To hold the user to the application they face strict deadline. In some cases they cannot get enough time to tune the product before introducing it to the market. Open source applications do not face this kind of problems. It comes to the market when it becomes ready enough for service.

So if someone thinks about the existence and future development of his application, he needs to develop it in the open source platform. Otherwise he has to work for it continuously to meet the competition.

3.3.3. Open source programming languages

Open source programming language provides the source code and the compiler for free. It’s mainly developed for non-commercial purpose. These languages are released with open source license. There are many open-source programming languages available in the market such as Java, PHP, Python etc. If we consider the popularity of the programming language we can see that, C, the mother language still keeps its popularity in the first position and then comes java, PHP, JavaScript, C++, python (Programming language www… 2013).

- **Java** has become popular because of its platform independence. That means it has huge acceptance in both desktop and mobile software development (Giacaman & Sinnen 2013). Java codes are run by the java virtual machine. It is not dependent on the architecture of the hardware. It has a big library with different types of classes.

- **PHP** is very popular for web application development. It is a server-site scripting language. It means the code runs the server and gives the output from the server to the user. It needs a server to run code. It can embed with HTML directly and without any necessity of processing unit.

- **Python** is newly developed object oriented programming language. Syntaxes of this language are easier than other programming language. It can run any operating system like windows, mac and linux. It also can run in .NET and java virtual machine.

If we consider the web development, PHP will come in the first position (Programming language www… 2013). But for developing CFD web application, we need to consider the libraries available for numerical calculations. For scientific application development python has been used widely because of its libraries for numerical calculation. As the objective of the study is not to develop application in different platforms like mobile devices, we didn’t consider Java for CFD application development.

3.3.4. Importance of web development framework

Web development framework gives a good pathway to developers in web development. When one builds an application from scratch, it is very time consuming and repetitive. But web development framework does that for them. So it has become a beautiful package for the developers rather than a pile of the same works(Plekhanova 2009). Key component of a web development framework is the programming language used in the framework for developing application. One programming language could have many frameworks. Software developers use a specific framework depending on their necessity or the kind of assistance expected from it. It reduces coding pressure on the developer.

Chapter - 4
Experimental Design

4.1. Python as a programming language

The uses of high level programming languages for mathematical calculation and numerical analysis are very common to the scientific application developer. In the high level programming languages the instruction architecture have changed from the machine code or assembly languages and need a compiler. Whereas in the low level programming language, the programming code can be converted in the machine code without any use of interpreter or compiler. This kind of languages runs very fast with less memory usage. But the high level programming languages are easier to understand and use. It also has visualization tools and different numerical method that are very important for CFD application development(Chudoba et al. 2013). Different high level programming languages i.e. Maple, Matlab, Octave, R and S+ are used in scientific application development. For the open source community python comes with its rich numerical libraries such as NumPy, SciPy etc. (Chudoba et al. 2013). The flexibility of using python scripting language carries more advantages for the developer (Langtangen 2006). It provides the opportunity to define the class by importing the library into the language that makes them more productive in calculation. Object visualization and dynamic control of state changes becomes easier to python developer by using extended attribute called model-view-controller (MVC) design pattern. For those effective benefits python have used as a programming language for development.

4.2.Web development in python

Web development in python has become a major issue with increasing amount of interaction between the user and the web. In the early times users could only see the static pages of the websites. They didn’t have the capability to generate or modify any contents of the websites. But now-a-days with different functionality of the websites users get the freedom to manage, modify and output their own interest in the web.

In a web application web server has a vital role and has changed with time. Previously browsers used to look for a specific static file in the server to display. Now the server follows the instructions given through a specific URL and generates a dynamic file, which is then showed by the browser. This has become a common infrastructure for a dynamic website. To generate file with a python code, HTTP server needs to understand the code. Usually the servers are written in C or C++ language so they cannot understand the python code. To make them understand it a common and accepted interface named CGI(Common Gateway Interface) (How to www… 2014) is used. Writing CGI script for complex programming and user interface was not so easy. Then came WSGI (Web Server Gateway Interface). If a programmer chooses a framework that supports WSGI, he doesn't need to get contract with it. When the code deploys, it is wrapped by WSGI. The server understands the code and executes it. In this case a web development framework which supports WSGI is necessary.

4.3. Web development framework

Web development framework is a collection of code written by a specific programming language that helps the developer write web application with that language. Python have many web development frameworks such as Django, Grok, TerboGear, Web2Py etc.

- Django was built for quick application development. It doesn't support repeatation of code rather it reuses them . It mainly focuses on doing things automatically without any support from the developer. It has an admin site which makes the database management very easy.
- Grok also have the philosophy of not repeating code and it gives emphasis on the convention of configuration. It was developed on Zope toolkit technology.
- Web2Py is like an all-in-one package. The web interface provided by the package supports almost everything including deployment and testing. It can run from a USB drive and installation is not required to execute. It has a web based IDE like IPython notebook.
- TerboGear has many WSGI components. It has good features to extend application for further development.

From the above frameworks Django is quite popular than the others because of its easy-to-use properties. It has a good documentation that attracts everyone to learn. It also has WSGI compatibility. So Django have been selected as a framework for python web development.

4.3.1. Django

In early ages, web development was limited only to editing HTML. CGI makes the web dynamic by generating HTML following the programming instruction. But in CGI script developer had to write so many repetitive codes. (Holovaty & Kaplan-Moss). PHP solved this problem but it was weak in security. Then the Django came with the solution for the security issue and repetition was no longer necessary. It follows the MVC design pattern but takes a slightly different approach. Django framework mainly developed on four parts. They are model.py, views.py, urls.py and templates.

Models.py controls the database file, views.py controls the business logic, urls.py controls the URL pattern and in the templates folder programmer saves the template files.

4.3.2. Setting up django in computer

For setting up django few tools have been used like easy-install, virtualenv etc. But at first python needs to be installed in the system. Python 2.7 is more compatible with Django rather than newer version like python 3.x. So python 2.7 is installed for this project. Easy install is a python module, which is available in the setuptool package and helps with downloading and installing different python packages. After installing python setuptools for using easy-install script, virtualenv is installed. Virtualenv is a structure, which we can use to separate the python application from the system. So the python stays in its own individual environment, which has its own packages and path. So everything installed in the virtual environment comes from its local path, not from the system. If the setup occurs in the system, there is a chance of clash within different versions. But by using virtual environment we can assign a specific folder to compact all the codes needed for the packages. So the package becomes individually independent. The following is used install virtualenv by easy-install in the terminal.

```
sudo easy_install virtualenv
```

Here sudo means ‘super user do’. It means this command is running as a super user. In an operating system many commands do not run for a general user. Running code as a super-user solves the problem and executes the code. After installing virtualenv one folder has been made by the virtualenv where django will be installed. For making a folder using virtualenv following command have used.

```
virtualenv –-no-site-packages django-sim-final
```

Here ‘--no-site-packages’ command is used for a specific interest. It restricts copying any python or other module from the system. So both python and django in this folder start from scratch without copying any file. A folder call django-sim-final has been created. This name was chosen without any specific cause. The user can define it by any other name. In this folder another folder have found called bin. Virtual environment needs to be activated from this folder. To activate virtual environment, following command is used.

```
source dijango-sim-final/bin/activate
```

The code just uses the path of the bin folder to activate virtual environment. It may be different depending on the path of the bin folder. It actually runs the root system from the folder rather than the actual root of the computer. By running this code ‘django-sim-fianl’ will be shown in the left corner under parenthesis. It means system is running from the virtual environment. So now any installation in this packages will happen in this environment. If the user goes to django-sim-final folder he will find bin, include, lib folders. This structure is the same as the root folder of a system. Django has been installed in the virtual environment. For installing Django, sudo command have not used because the folder acts as a root folder. The code used for installing Django:

```
easy_install Django
```

The process done by the easy_install command is as follows:

- First it goes to the python package index website and look for the package called Django.
- Then it goes to the Django project site and got the detail about the Django
- Then it looks for the current version for Django and downloads the zip file.

- After downloading it installs the Django in the virtual current environment.

In the bin folder of the virtual environment, django-admin.py file has appeared. By the following ‘startproject’ command, django-admin.py has made a new project for django.

```
django-admin.py startproject django_sim_final
```

There is no convention for naming the project. It can be named according to the programmer’s choice. Project has become ready for development.

4.3.3. MTV

Like MVC pattern Django has its own design pattern called MTV design pattern. The main interest for the MVC and the MTV pattern is same. The principle of the MTV(Model, Template and View) design pattern is to separate the data access logic, business logic and presentation logic. Another important topic is URL pattern. These four types of codes are written in four separate files. The advantage of having four separate files is significant. When the programmer needs to change one part of the codes he can just change a single file (Holovaty & Kaplan-Moss 2009). He doesn’t need to search the whole code to find what needs to change. Suppose the programmer needs to change the template of the website. He can go to the template folder directly without thinking about other files. On the other hand, if he wants to change the database structure of the application he just go to the model.py file for database management. Its saves a lot of time for the programmer and the coding becomes clear and understandable to others.

4.3.4. Model

Model is a specific bunch of code written for database management. It could be found in the model.py file. Naming it model.py is not necessary but proper connection with other files is important. The model structure is also called data access layer. Many web applications access data from the database for visualizing output. In a database driven website, it connect to the database, takes the required data, shows the data to the website and stops the connection with the database (Holovaty & Kaplan-Moss 2009). For the primary development of CFD web application it is not necessary for the simulation to make the database. So data access layer is not used in the application.

4.3.5. Template

Template contains the presentation of the web application. Previously the code was written in the python code. But now Django has separated it. Web design and development are two totally different disciplines. The web designers mainly do template design by changing HTML or CSS code.

Now by separating templates from raw python code it becomes easier for the web designer to change the look and fill of the website without understanding python code. On the other hand, in the development environment both the application developer and the designer need to work simultaneously. If the codes are written the same place, they might cause conflict because the web developer and the designer could change same file at the same time. In this application HTML and CSS have used for template design.

HTML

Programmers are using HTML coding from long ago. HTML stands for hypertext mark-up language. HTML files have the extension of .html or .htm. This extension does not need to be changed to use in Django. HTML files normally have two parts. One part lies under title tag and another under body tag. The part under body tag is the main part in the html file. General convention is to make a templates folder in the projects and let Django know its position of the templates folder by inserting the path in the settings.py file.

CSS

CSS stands for Cascading Style Sheet. Main objective of the CSS coding is to change the layout of the webpage with less effort. It means that it uses lesser codes for specific output, which loads the webpage faster. It stops doing repetitive works. Changing a single line in the CSS file can change the whole webpage with similar types of structure. To attach CSS characteristics with HTML, it makes class or ID. By calling CSS class or id in a single tag, the tag gets the specific characteristics defined by CSS class or id.

4.3.6. View

View part of Django, which is the business logic part, can be found in the views.py file. Normally, different methods for web application are defined in this part. In the view file functions return the HttpResponse which generates the .html file.

4.3.7. Settings.py

Settings.py file in Django is a very important file. It contains all the configuration of the Django framework. As it has all the sensitive information like database password, it should be secured. So the permission should be limited for this file. It also has another important part for web application, which is TEMPLATE_DIR. It shows the path of the templates for websites. When it needs to find a template, it looks for the .html file in this path.

4.3.8. URL pattern

The entire URL for the web application has defined in a variable called urlpattern that is available in a file named urls.py. In the urlpattern, URL is defined in the urlpattern by giving the URL first and name of the function that is needed for the URL in the second position. Functions are called from views.py file. There are some roles for specifying URL. By these roles, programmer can make dynamic URL where one URL works for same type of many webpages only by changing the value. The URL generated pages will be different depending on the dynamic urlpattern.

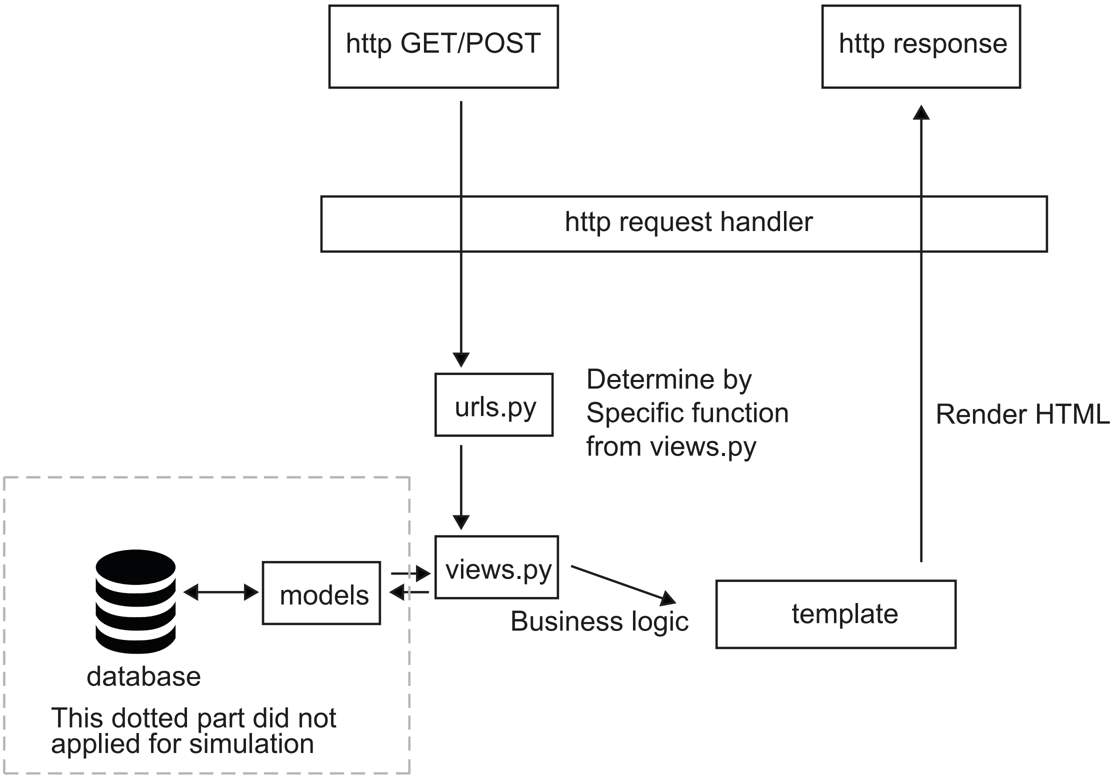

Figure 1: Django web development framework structure

4.4.Python Libraries for CFD

For CFD simulation two types of libraries are needed. One is for numerical calculation and the other for plotting the result and visualizing the image. Different types of libraries are available for python. For example Matplotlib, NumPy, Plotly, SciPy etc.

4.4.1. NumPy

NumPy was developed for doing numerical calculation by python. There are mainly two fundamental objects in NumPy. One is n-functional array and another is universal function. N-dimensional is a very significant feature for CFD simulation. It is a collection of similar items indexed by N integer. Arrays have two main parts. One is the shape of the array, which indicates to the point where the index could vary. Other is the characteristics of every individual objects because they are same type of dataset and occupies same block of memory (Oliphant 2006).

4.4.2. Plotly

Plotly is a tool for data analysis and visualization. It supports most of the data format. By using custom function over data, output can be attained. It is possible to import another python package like numpy, SciPy etc in Plotly. The output could be exported at different file format like PNG, SVG, PDF and EPS. To adjust the output layout, the user have full control over it. It works with python through API. It could raise complexity for Django and it is not designed for complex data analysis.

4.4.3. SciPy

SciPy is a stack of packages for scientific computing. It contains different packages like NumPy, matplotlib, IPython etc and also contains its own library. Its main interest is to fulfill the scientific need by using python. The use of SciPy is normally shown in IPython. By installing, it could be used in any development environment.

4.4.4. Matplotlib

Matplotlib is a 2D plotting library and have similar functionality like MATLAB. It can be used in python script, ipython shell, web application server etc. By using matplotlib plotting has become easy. It needs just a few lines of codes. Matplotlib follows the object oriented programming style and is compatible with large application development. Most programmers use command line interface, which provides a pop up window for displaying plotted data. Matplotlib mainly have three different parts (Barrett et al. 2005).

- Matplotlib interface – A command line interface of a set of functions for plotting.
- The frontends or Matplotlib api – A set of classes that manages figure, line, plots etc. as output.
- The backends – It transports the frontend output to hardcopy like PNG, JPEG, PDF or SVG format. In case of some complex rendering, code is written in C/C++ for getting better output.

For visualizing output from CFD numerical calculation Matplotlib is used in Django.

4.4.5. Setting up NumPy and Matplotlib in Django

For setting up NumPy and matplotlib in Django, virtual environment has been activated through terminal. After activating virtual environment, NumPy and matplotlib have tried to install by using easy_install command. But an error occurred during installation. Afterwards pip has been used to install NumPy and matplotlib. It installed without any error. Pip is a package manager for python packages. It installs and maintains packages. It has a command line interface, which is very easy to use. For example, it uses only one line of code to install packages.

4.5.CFD simulation using Navier-stokes equation (cavity flow)

4.5.1. Navier-stokes equation

Navier-stokes equation discusses the fluid flow, which is the result of Newton's 2nd low over fluid motion. This equation is used for understanding fluid flow in ocean current, flow around the wings, flow in pipe etc. Navier-stokes equation is a simplified explanation of fluid flow that has helped to design different structures using CFD.

Navier-stokes equation for incompressible fluid flow (Landau 1987):

$$\nabla . \vec{u} = 0 \qquad (1)$$

$$\frac{d\vec{u}}{dt} + (\vec{u}.\nabla)\vec{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \vec{u} \qquad (2)$$

Equation (1) represents the conservation of mass at constant density. Equation (2) represents the conservation of momentum. Here $\vec{u}$ is the velocity, P is the pressure, ρ is the density of the fluid and ν is the kinematic viscosity.

If we consider the equation (2), $\frac{d\vec{u}}{dt}$ is the unsteady flow with time, $(\vec{u}.\nabla)\vec{u}$ is the convective term including first order derivative, $\frac{\nabla p}{\rho}$ is the gradient of the pressure with density and $\nu\nabla^2\vec{u}$ is the diffusive term including second order derivative.

Equation (2) is a simplified form. To use this equation in CFD, a structure of PDEs is necessary. One numerical methods needs to develop that satisfies the mathematical properties of PDEs for example diffusion terms have second order derivative.

Physical phenomena needs to consider that could happen in this simplified model including diffusion and convection. For example we need to consider the initial condition or the boundary condition.

4.5.2. The finite difference method

Numerical method that has used for the equation is finite difference method. This method uses Tylor series expansion to get the approximation of the derivative depending on the variable of the partial differential equation. First step of the method is discretization. It discretizes the numerical domain by using a Cartesian coordinate system in the grid. It means the gridlines of same family will never intersect and gridlines of two different families will intersect only one time. Each node of the grid has been identified by indexes like i and j etc. The main mechanism of indexing is to name the neighboring nodes by increasing and decreasing the index. For example i+1, i-1 etc. Every node has unknown value depending on the variable that has been considered and the value of the nodes also depends on the neighboring nodes.

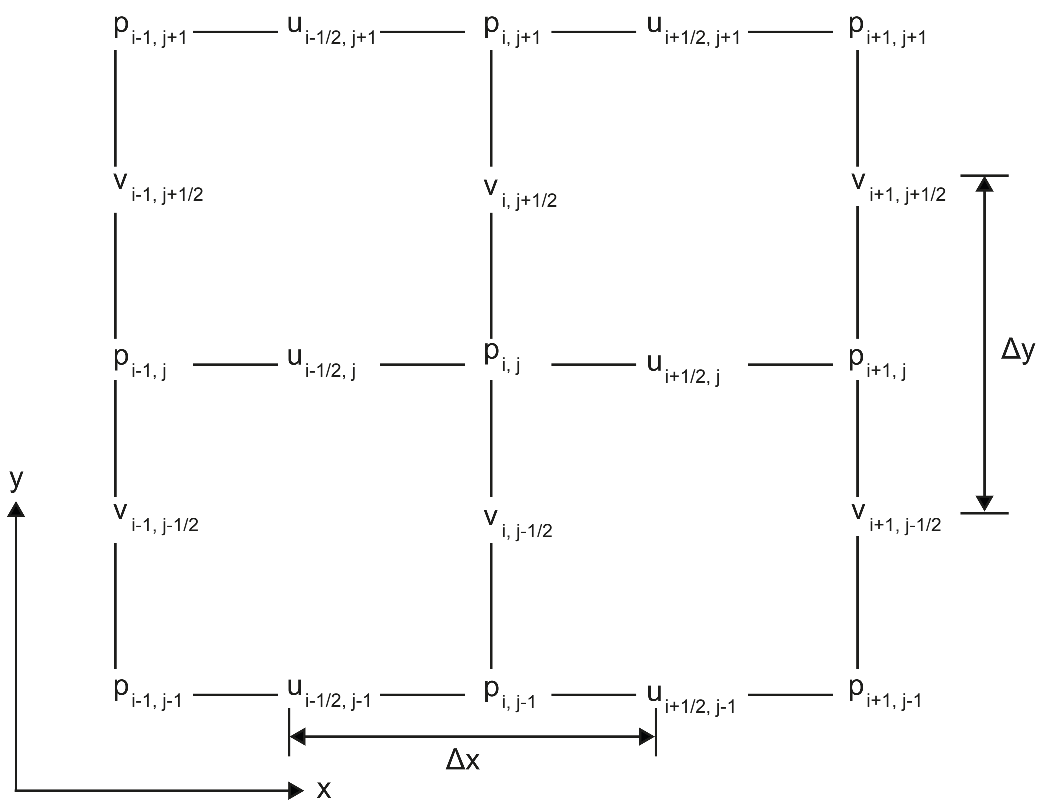

Figure 2: Cartesian grid

In a finite difference method three forms are normally used depending on the variable. They are forward difference, backward difference and central difference (Non-linear convection www… 2014). When the difference between two nodes (Δx) decreases, the result become more approximate. If we consider x is the length of the grid and the difference between two nodes (Δx) are uniform and the flow velocity is u then the differences with respect to space will be calculated as follows:

Backward difference: $\frac{du}{dx} \cong \frac{u_i - u_{i-1}}{\Delta x}$ (3)

Forward difference: $\frac{du}{dx} \cong \frac{u_{i+1} - u_i}{\Delta x}$ (4)

Central difference: $\frac{du}{dx} \cong \frac{u_{i+1} - u_{i-1}}{2\Delta x}$ (5)

By using Taylor series expansion approximate value for $\frac{du}{dx}$ need to be achieved. If we consider the Taylor series expansion of $u(x)$ for the point x_i then equation will be as follows:

$$u(x) = u(x_i) + (x - x_i)\frac{du}{dx}\bigg|_i + \frac{(x-x_i)^2}{2!}\frac{d^2u}{dx^2}\bigg|_i + \cdots + \frac{(x-x_i)^n}{n!}\frac{d^nu}{dx^n}\bigg|_i + \cdots \quad (6)$$

Here if x has replaced by x_{i+1} and try to find the equation for first order derivative $\frac{du}{dx}$ then the equation:

$$\frac{du}{dx}\bigg|_i = \frac{u_{i+1}-u_i}{x_{i+1}-x_i} - \frac{(x_{i+1}-x_i)^2}{2}\frac{d^2u}{dx^2}\bigg|_i - \cdots - \frac{(x_{i+1}-x_i)^n}{n!}\frac{d^nu}{dx^n}\bigg|_i - \cdots \quad (7)$$

If $(x_{i+1} - x_i)$ is very small, it can neglected with all the higher order term. Therefore the forward difference approximation will be as follows:

$$\frac{du}{dx}\bigg|_i = \frac{u_{i+1}-u_i}{x_{i+1}-x_i} \quad (8)$$

The neglected part has called truncation error. For the condition, Δx $(x_{i+1} - x_i) \longrightarrow 0$, the error will be zero and the result will be more approximate. Backward difference and central difference can be calculated by the same way. A central difference can be obtained by combining forward difference and backward difference. This happens because the first order derivative has cancelled out after combining forward and backward difference.

To calculate the velocity for specific time, space-time discretization needs to be considered. For this purpose, time need to be considered with space. For example if the initial condition, $u(x)$ have given at time t_0 which could be consider as $u_0(x)$. For a given time n, the changes of velocity can be calculated which would be $u_n(x)$.

One easy equation can explain how velocity can be achieved in different time steps.

$$\frac{du}{dt} + u\frac{du}{dx} = 0 \qquad (9)$$

After discretizing the equation:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \qquad (10)$$

In this equation changes of time and space have written as Δt and Δx for simplicity. In velocity (u_i^{n+1}), two indexes are available n and i. n represents time index and i represent space index. Here only the unknown value is u_i^{n+1}. Other values have found from the initial condition. For getting value for u_i^{n+1} the equation would be as follows:

$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) \qquad (11)$$

After getting the value for u_i^{n+1}, it can be used for getting another velocity, $u_i^{n+2}$and so on by applying the same equation. For achieving velocity in every time steps, a loop needs to run through computational process.

However, this equation results the velocity in one dimension (1D). To convert this equation in two-dimensional form, 2D grid has been applied. In two dimensional (2D) grid, x represents the space in x-axis and y represents the space in y-axis. When partial derivative is calculated with respect to x, y will be constant. So every value for x will be calculated by keeping the y value in the same position.

Therefore the Equation 11 in 2D form will be as follows:

$$u_{i,j}^{n+1} = u_{i,j}^{n} - u_{i,j}^{n} \frac{\Delta t}{\Delta x} (u_{i,j}^{n} - u_{i-1,j}^{n}) \quad (12)$$

Here, j is the indexes for y value and is constant for the equation. Velocity for every time step in specific position can be calculated in 2D by this way.

4.5.3. Finite difference method in Navier-stokes equation

Navier-stokes equation from equation (2):

$$\frac{d\vec{u}}{dt} + (\vec{u}.\nabla)\vec{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \vec{u}$$

For applying this equation in 2D grid, two velocities have considered. In x-axis the velocity is u and in y-axis the velocity is v. For getting velocity u in different space and time equation (2) could be represented as follows:

$$\frac{du}{dt} + u\frac{du}{dx} + v\frac{du}{dy} = -\frac{1}{\rho}\frac{dp}{dx} + \nu(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}) \quad (13)$$

For getting velocity v in different space and time equation (2) could be represented as follows:

$$\frac{dv}{dt} + u\frac{dv}{dx} + v\frac{dv}{dy} = -\frac{1}{\rho}\frac{dp}{dy} + \nu(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}) \quad (14)$$

For getting pressure in different time steps Poission equation have applied.

$$\frac{d^2p}{dx^2} + \frac{d^2p}{dy^2} = -\rho(\frac{du}{dx}\frac{du}{dx} + 2\frac{du}{dy}\frac{dv}{dx} + \frac{dv}{dy}\frac{dv}{dy}) \quad (15)$$

By discretizing equation (13) velocity $u_{i,j}^{n+1}$ can be achieved in n+1 time step.

$$u_{i,j}^{n+1} = u_{i,j}^{n} - u_{i,j}^{n} \frac{\Delta t}{\Delta x} \left(u_{i,j}^{n} - u_{i-1,j}^{n}\right) - v_{i,j}^{n} \frac{\Delta t}{\Delta y} \left(u_{i,j}^{n} - u_{i,j-1}^{n}\right) - \frac{\Delta t}{\rho 2\Delta x} \left(p_{i+1,j}^{n} - p_{i-1,j}^{n}\right) + \nu \left(\frac{\Delta t}{\Delta x^2} \left(u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}\right) + \frac{\Delta t}{\Delta y^2} \left(u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}\right)\right) \quad (16)$$

By discretizing equation (14) velocity $v_{i,j}^{n+1}$ can be achieved in n+1 time step.

$$v_{i,j}^{n+1} = v_{i,j}^{n} - u_{i,j}^{n}\frac{\Delta t}{\Delta x}\left(v_{i,j}^{n} - v_{i-1,j}^{n}\right) - v_{i,j}^{n}\frac{\Delta t}{\Delta y}\left(v_{i,j}^{n} - v_{i,j-1}^{n}\right) - \frac{\Delta t}{\rho 2\Delta y}\left(p_{i,j+1}^{n} - p_{i,j-1}^{n}\right) + \nu\left(\frac{\Delta t}{\Delta x^2}\left(v_{i+1,j}^{n} - 2v_{i,j}^{n} + v_{i-1,j}^{n}\right) + \frac{\Delta t}{\Delta y^2}\left(v_{i,j+1}^{n} - 2v_{i,j}^{n} + v_{i,j-1}^{n}\right)\right) \quad (17)$$

Pressure $p_{i,j}^{n}$ can be achieved by discretizing equation (15).

$$p_{i,j}^{n} = \frac{(p_{i+1,j}^{n}+p_{i-1,j}^{n})\Delta y^2+(p_{i,j+1}^{n}+p_{i,j-1}^{n})\Delta x^2}{2(\Delta x^2+\Delta y^2)} - \frac{\rho\Delta x^2\Delta y^2}{2(\Delta x^2+\Delta y^2)} \times \left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1}-v_{i,j-1}}{2\Delta y}\right) - \frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x} - 2\frac{u_{i,j+1}-u_{i,j-1}}{2\Delta y}\frac{v_{i+1,j}-v_{i-1,j}}{2\Delta x} - \frac{v_{i,j+1}-v_{i,j-1}}{2\Delta y}\frac{v_{i,j+1}-v_{i,j-1}}{2\Delta y}\right] \quad (18)$$

4.5.4. Naiver-stokes equation in Django

To apply Naiver-stokes equation in Django the following points need to be considered:

4.5.4.1.URL pattern

In the url-pattern, one url has been defined for the user interface where user can input data for simulation. Another url has been defined to generate the result as an image.

4.5.4.2. User interface

Web application has a user interface where the user can enter data. Here nx represent number of point in x-axis and ny represents number of point in y-axis. nt represent number of timesteps. nit represent number of iteration. Other data has kept constant in the simulation function. User interface is an html file, which is available in the template folder. After entering data when the user presses the run button, the data is redirected to the view file by GET method. GET method is a process in the html form that can grab the data for further analysis.

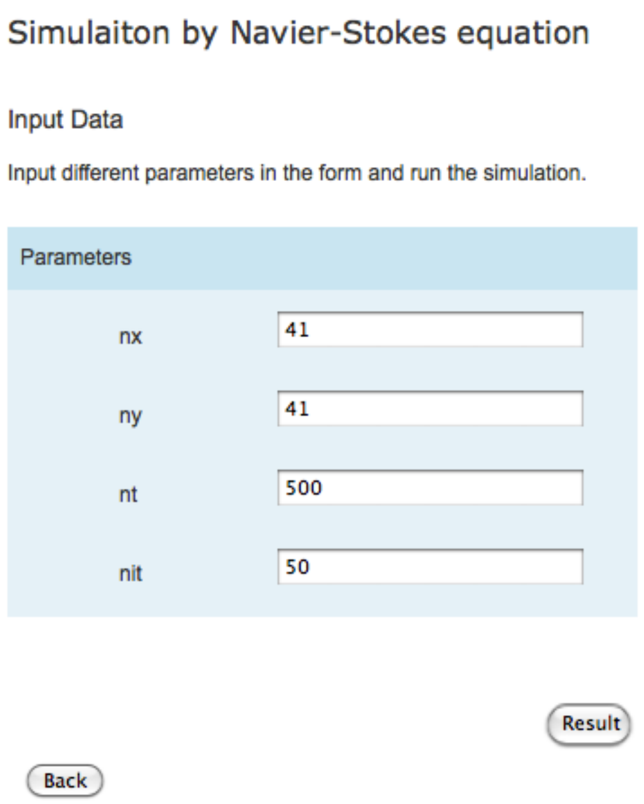

Figure 3: User interface to input data for cavity flow simulation

4.5.4.3.Views.py

Views.py file has all the functionality for naiver-stokes equation. When the get method brings the data in the view file, they enter into different variable, which could be used later in different functionality. Pressure equation can be divided into tow functions for simplicity. Other equations are written in one function. To run the simulation and to plot the pressure and the velocity different libraries have been imported. To render the html file for a specific URL 'render' has been imported from django.shortcuts. HttpResponse has been imported from django.http to produce the http output as an image. Axes3D, cm, FigureCanvasAgg, Figure, plt have been imported from matplotlib for plotting output in an image. Numpy have imported for defining array function.

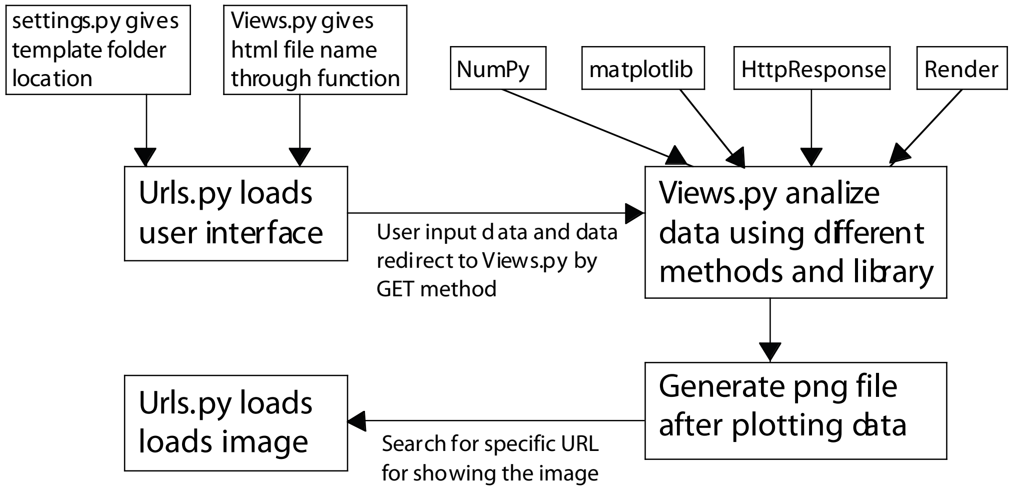

Figure 4: Processes in Django for simulation

Figure 4 shows the workflow in the Django for running simulation. Here views.py file have got the most importance. It imported all the libraries and run all the function for the simulation. urls.py loads dynamic web pages for the UI and the image by following the methods declared in the views.py file. settings.py file control all the settings in the Django framework like project settings, application settings, template path, authentication, administration etc. It provides the path of templates for this simulation.

Total process is interconnected with each other and has a clear code structure with the help of Django. This process also can run other simulation by discretizing the model equation and by making the CFD solver for a specific CFD model.

4.6. Developing web application for sedimentation tank

4.6.1. Different models for multiphase flow simulation

Main mechanism of sedimentation tank is to separate the phases to improve the water quality (Ghawi & Kriš). Different techniques have been applied to improve the efficiency of the tank depending on different factors related to the performance of the sedimentation tank. The physical condition of the flow field and the suspended solid in the sedimentation tank can be characterized as a multiphase flow. It can be explained by two different approaches i.e. Euler-Euler approach and Euler-Lagrange approach. Eulerian method studies the fluid flow by considering a specific region and studying the amount of particles going through that region in a specific time interval. On the other hand legrangian method tries to study trajectories of each particle in the fluid flow (Tarpagkou & Pantokratoras 2013). Because of its attention over individual particle in a fluid flow and particle settling in sedimentation tank, legrangian method is commonly used for studying multiphase flow. In a lagrangian method, primary phase (water) and the secondary phase (particle) may not interact with each other. Particles do not show any coupling effect with the water. But it happens when the concentration of the secondary phase in the tank is very low. If more than 10%-12% concentration of discrete phase (particle) is found in the wastewater, it cannot be explained by the lagrangian method (De Clercq et al. 2005). Eulerian method is used for this purpose. Lagregian method could be used in those wastewater treatment plant that supply potable water. For understanding industrial wastewater with high concentration of waste, eulerian method is used in some cases. Because of having no interaction between the two phases in lagragian approach, conservation equation has been solved separately for each phase. In this case particle size and volume have been specified in the initial condition (Fan et al. 2007). Whereas in eulerian approach concentration of suspended solid has been solved by one species transport equation and the primary phase has been solved as a single phase.

Fluid flow could be turbulent in a sedimentation tank. In this case, time-average information has been gathered by solving conservation equation. It retains an additional part in the conservation equation, which is the conservation of mass and momentum by the turbulence flow. The $k - \varepsilon$ turbulence model can explain this part.

It explains the transportation of the particle from the inlet to the deposition point governed by the turbulence flow (Kahane et al. 2002). This model makes a link with the Renolds stress. Other models are also available for turbulence flow such as eddy velocity model, renolds stress equation model (RSM). In some cases, another modified model called RNG $k - \varepsilon$ model originating from the $k - \varepsilon$ turbulence model has been applied (Tarpagkou & Pantokratoras 2013). This model is more applicable than the standard one and more compatible for eddy. It considers less renolds number effect.

In sedimentation tank, many researchers study about another important issue called the dead zone. Dead zones are those areas where circulation occurs in the sedimentation tank and they have significant effect in the sedimentation process. They reduce deposition spaces for the suspended solid. To reduce the dead zone and to increase the efficiency of the sedimentation tank, baffle has been introduced in the sedimentation tank. Proper position of baffle can divide the circulation zone and increase the efficiency of the sedimentation tank by providing more space for deposition than before (Razmi et al. 2009). Uniform distribution of velocity could be ensured by using baffle that can help to deposit particle in a short time (Tamayol et al. 2008). To understand the effect of introduction of baffle in a flow field of a sedimentation tank, Navier-Stokes equation with Renolds stress have used by researchers as a time-averaged flow equation. $k - \varepsilon$ model have introduced as turbulence model to study the turbulence flow in the sedimentation tank (Tarpagkou & Pantokratoras 2013).

4.6.2 Simulation by desktop application

For CFD simulation, different desktop application has improved remarkably. To understand the flow behavior and the geometry of the sedimentation tank it is necessary to run some simulation through the available popular desktop application. It also can provide necessary guideline about the process to implement CFD simulation in a web platform for sedimentation tank.

One wastewater treatment plant was visited, situated in Drøbak, which has a sedimentation tank with 32m in length, 7m in width and 3.6m in depth. This sedimentation tank can be characterized as secondary sedimentation tank, because it used coagulation process to settling particles. It is a rectangular sedimentation tank without any geometrical modification. It has two inlets and three long outlets launder. It is not include any baffle also. In this case, for the velocity in the sedimentation tank, particles do not get proper time sedimentation. Also a dead zone could cover some part of the sedimentation tank in the inlet area because of circular current. It could reduce the effective area of the sedimentation tank and can reduce the efficiency of that unit. There is another chance of mixing the particle with the water again because of dead zone. So it is necessary to reduce the circulating zone in the sedimentation tank to get more efficiency. Introducing baffle is an effective solution to reduce circulating zone in the sedimentation tank. But introducing baffle also could cause problem if velocity is high in the inlet zone and circulation could pass the baffle and can reduce the effective sedimentation area. Another effective solution could be changing the shape of the sedimentation tank. If the depth of the sedimentation tank has increased in the inlet area it can effectively reduce the circulation zone and can show better efficiency.

Two simulations have run by modifying the geometry of the sedimentation tank. One simulation has run by introducing slope and baffle in the sedimentation tank and another simulation has run by introducing double baffle in the sedimentation tank. Fluent application has been used for the simulation process. Here the inlet was kept rectangular in shape and the dimension has kept 0.4*0.4 m in size. Outlet launder in the middle position has kept longer than other two by following the actual structure of the sedimentation tank.

4.6.3. Web structure for CFD simulation

Different CFD desktop applications have used to optimize sedimentation tank by many researchers. Both commercial and desktop applications are available for CFD simulation. For running a simulation in a web platform need a proper architecture about the total process. Mainly, the web application can be divided into two parts:

1. Front-end
2. Back-end

Back-end part needs to explain before front-end because the core part of the web application has designed to develop in the back-end part. Front-end part is mainly developed depending on the back-end functionality.

4.6.3.1 Back-end structure

Back-end part of the web application is the data analysis part. It will get the data form the front-end part, analysis the data and provide the result to the front-end. In the front-end part user will provide some data about the initial condition and the boundary condition of their desired sedimentation tank.

Getting data form the database

Django database structure has been used to get data from the front-end part. When the user provides data in the user interface, the data will save in the database first. Afterwards, back-end part will grab this data from the database for analysis. Database was not used in the previous simulation for cavity flow calculation. However, fluids in a sedimentation tank have two phases. One is sewage and the other is solid particle. So multiphase flow simulation is necessary to consider for studying sedimentation tank. To understand the flow pattern and volume fraction sewage and solid particle more clearly, 3D (three dimensional) simulation is necessary. In this case, the simulation will take longer time to be completed. Another important factor is to generate the mesh file for the sedimentation. This process also consumes time. All the data provided from the user will not be used at a time. First the back-end part gathers data for mesh generation and generate the CAD geometry and make the mesh file. Then it will take the data for CFD model. This data will choose CFD solver and other necessary model for simulation. All the data need to save in the database for diversified use by the back-end part of the application.

Another useful application of using database is to save the data for the user. To use the web application, users will be asked to be registered in the process. After registration, user can reach into his workflow. When user will run one simulation, data would be saved as a project. If user needs to run another simulation using different mesh file and want to provide same data, they could take the data form the previously saved project. Also the data provided for present simulation can be used for future experiment.

Data analysis for CFD simulation

Provided data from the user interface will be analyzed to complete some tasks:

- CAD geometry generation for the sedimentation tank.
- Mesh structure generation from the CAD file.
- Running simulation by using CFD model and solver.
- Image generation as an output of the CFD simulation.

In the previous simulation all the functions were available in the views.py file of Django framework. To run the simulation inside the Django framework, specific CFD model and solver need to run inside the framework. But there are some difficulties of importing solver in the Django framework.

- As most of the CFD applications are desktop application, coding structure is different from the web application. In a desktop application all the codes are integrated with each other. It is difficult to sort out independent code for a specific CFD model.
- Applications for generating mesh file are like a single package. It is very difficult to separate specific code for specific mesh generation. The same problem arises with the software, which visualize output result for the simulation.

Therefore, a total simulation package which contains all the functionality for running simulation could be used in the web. As the objective of the study is to develop a CFD web application, it is necessary to make a connection between the web application and the CFD applications. One similar kind of work has done by Altair (CFDCalc simple www… 2013). They have used python script and shell script to run solvers in the cloud and python script to run web interface. Another similar work have done by using OpenFOAM in the back-end (Sempolinski et al. 2012). Python scripts have used to interact between front-end and back-end. So by using python scripting language and OpenFOAM, one web platform can be built which can study sedimentation tank. Here, web development framework Django can be use for user interface and to manage database.

On the other hand, there are some limitations of developing CFD numerical codes for simulations. Different CFD models are available for studying fluid flow in sedimentation tank. To use those models in CFD simulation, discretization the model equation is the first process. Then the discretized equation needs to convert to a computational code. Then computational codes have analyzed and generate output images by plotting different parameters. To execute the total process some problem may occur in different part of the simulation process:

- Mesh generation:
  - For sedimentation tank, mesh generation is a big challenge. If the web application is developed from scratch, one mesh generation library needs to be created. Creating a mesh generation library for 3D simulation is complicated and beyond the objective of this study. Create a mesh by using available mesh library can save time and workload.
  - User verified mesh generation product would be safe for running simulation. It will be bug free also.
  - Applications, those have many users need to be modified in every version. To fulfill different requirements from different types of user, developer adds different features. This process makes the application more usable and user-friendly. So an existing popular mesh generation product should have these characteristics.
- Output visualization:
  - Another important issue is to visualize the output. For cavity flow visualization by naiver-stokes equation, matplotlib has been used to plot and visualize different parameter in image. Matplotlib is a 2D plotting library. However, visualization of a sedimentation tank in a 3D perspective can explain the flow field more elaborately.
- Numerical analysis:
  - Making a solver for a CFD model sometimes can cause problem to get an output. In this case, question arises about the accuracy of the codes. Therefore, one should use well-verified code rather than a untested code. It is important to use those solvers, which have well verified code.

- Different researchers have done their CFD project by using different commercial and open source CFD application. Different CFD application has many features to ensure the accuracy for CFD simulation and to explain the result more elaborately. Using those features can make the scientific work easier for the readers to understand.

From the above circumstances, applications those gives access to its code (open source) have decided to uses for the sedimentation tank.

Different open-source CFD applications:

There are some open source CFD applications available in the market. For example OpenFOAM, CAELinux, Elmer etc. However, they are desktop application. Most of the application developed for linux platform. Among them OpenFOAM is popular. It has large amounts of CFD model solvers (Standard Solvers www…). For multiphase flow simulation it has different solvers. By considering the physical condition of the sedimentation tank and the fluid flow, few solvers could be applied for studying the sedimentation tank. They are interFoam, LTSInterFoam, MRFInterFoam, multiphaseInterFoam, settlingFoam, twoPhaseEulerFoam etc (Standard Solvers www…). All of the solvers study two incompressible, isothermal, immiscible fluids. Here settlingFoam solver is closely related to settling tank where discussed about dispersed phase settlement.

Supporting application for OpenFOAM:

OpenFOAM is a CFD application for numerical analysis. To complete CFD simulation, it needs a mesh geometry, which could be provided for another application. After completing simulation it also need visualization software, which will show the output.

OpenFOAM has its own mesh generation part such as blockMesh, snappyHexMesh etc (Mesh generation www…). There are also some mesh generation software available for running simulation in OpenFOAM. For example, Gmsh, SALOME etc. Paraview is visualization software, which can read OpenFOAM output and can visualize the output file generated by OpenFOAM (Ahrens et al. 2005).

Integration of CFD application in web application

Three CFD open source applications have selected for studying sedimentation tank. They are SALOME, OpenFOAM and paraview. These applications are developed for linux platform. Ubuntu platform has been selected to run those applications. To run those applications from the python web interface one python script is necessary to run a shell script. And by running a shell script, application will be executed.

To run a shell script by python script different process can be used. One module in python called subprocess contains most upgraded features for controlling shell script (subprocess www… 2014). Besides there is also another module called os, which have similar functionality.

4.6.3.2. Front-end:

For multiphase flow simulations, computer needs to calculate large amount of numerical problems. This simulation is not possible to complete in a few moment. It could take either less than one hour or more than several hours. So it could be boring and time consuming for the user to sit in front of the monitor. To solve this problem, user authentication can be implemented in the web application. In the front end user interface, Django framework will be used. When user login to the application, one workflow for the user will be appeared. This workflow template will be generated from the predefined function available in the views.py file. When user put customized data in the user interface, it will save in the database through model.py. This data will be used for numerical analysis in the back-end simulation part.

In the primary data, there are sections for simulation data and the mesh data. User will have the option to customize the value for height, width and depth of the sedimentation tank. Initial condition and the boundary condition will be found in the default mesh options.

One mesh image for each mesh file will be available in the user interface and some information (initial conditions and boundary conditions) will be available with the image. This information will be fixed to reduce coding complexity. When the back end process will run, user will see busy sign in the user interface. When the simulation has finished, user will see the output image generated by the back-end process.

Output visualization:

When back-end part of the application will complete the total process, user will see the output image. One python script will grab the output image form the specific folder and show the image in the user interface. Urls.py will generate a dynamic URL for the specific user to show output image.

Chapter - 5
The way forward

In the web application, front-end part has completed but the back-end part has not finished yet. Major remaining part is to make the connection with the different application and to maintain different application form the front-end. After completing this task, the back-end of the web application will run following the structure below:

Back-end structure

Back-end simulation completes all processes from pre-processing (mesh generation) to post processing (image generation). Total procedure will be done in the server computer. Server computer need to have a linux platform. Because applications those will be used for running simulation have only developed for linux platform. Three applications (SALOME, OpenFOAM and Paraview) will run in this process. These applications are open source applications. User, who will use this website, will not have direct contact with the applications. Users only can see the web interface.

This process can be divided into three different parts:

1. Mesh generation
2. Running simulation
3. Output image

Mesh generation

The application will generate mesh depending on the primary data that have provided. The system also can have a data file, which contains default data for mesh generation. If the user does not provide any necessary data for mesh generation, that data will be provided from the data file.

To make the work done fast, some mesh file for rectangular sedimentation tank will be available in a specific folder. For those files some options will be avalilable in the user interface. User can choose one specific mesh with specific geometry. If a user provides a specific size (like height, width, depth etc) of a sedimentation tank, one python file will work behind it to put the specification in the mesh generation library. SALOME, which is a mesh generation software, will be used to generate all the mesh (Bergeaud & Tajchman 2007). The user will find many default mesh options, which will fulfill almost all types of inlet, outlet, boundary conditions and initial conditions. By choosing a default structured mesh users can save time and the total process will finish quicker than customized mesh generation process. In this case SALOME will not execute and the workflow goes to the second step for running the simulation.

For customized mesh generation, shell script will execute the application and python script will work to complete three different steps:

1. Grab customized data from the database and provide initial condition for the sedimentation tank
2. Generate the mesh file
3. Save the mesh file in a specific folder

Running simulation

Users will have the options about which model they want to run for the simulation. These options will be provided from the solver library and the other modules from OpenFOAM (Standard Solvers www…). There are many solvers available for multiphase flow simulation in the OpenFOAM solvers library. User will have the options to choose some of those solver and other CFD model for sedimentation tank simulation. For running simulation one python script will look for the specific mesh file in a specific folder. Then it looks for a specific solver and model in a OpenFOAM library. After getting this it will run the simulation. When the simulation will be completed and file will be generated as output.

Output image

For plotting data and output image, one application will be used called paraview. Paraview is an open source application, which can generate 3D image for simulation. One shell script will execute paraview and python script will show the path of the output file, which is generated by OpenFOAM. Paraview will run the file and save the output as image in a specific folder. Steps will be done in this part as follows:

1. Detect the path where OpenFOM output file have saved.

2. Execute paraview application
3. Take the output file and run in paraview.
4. Generate an output image after running the output file form OpenFOAM.
5. Save the output image in a specific folder.

Chapter - 6
Result

6.1. Cavity flow by naiver-stokes equation

Simulation structure has been established in the Django framework and codes for the simulation have written in python programming language. Simulation has run for 6 different time steps to see the changes of the velocity. As output, 6 files have generated with the resolution of 1100pixel*700pixel and with 100dpi(dot per inch). The size and resolution of the output file have declared at the views.py file in Django framework.

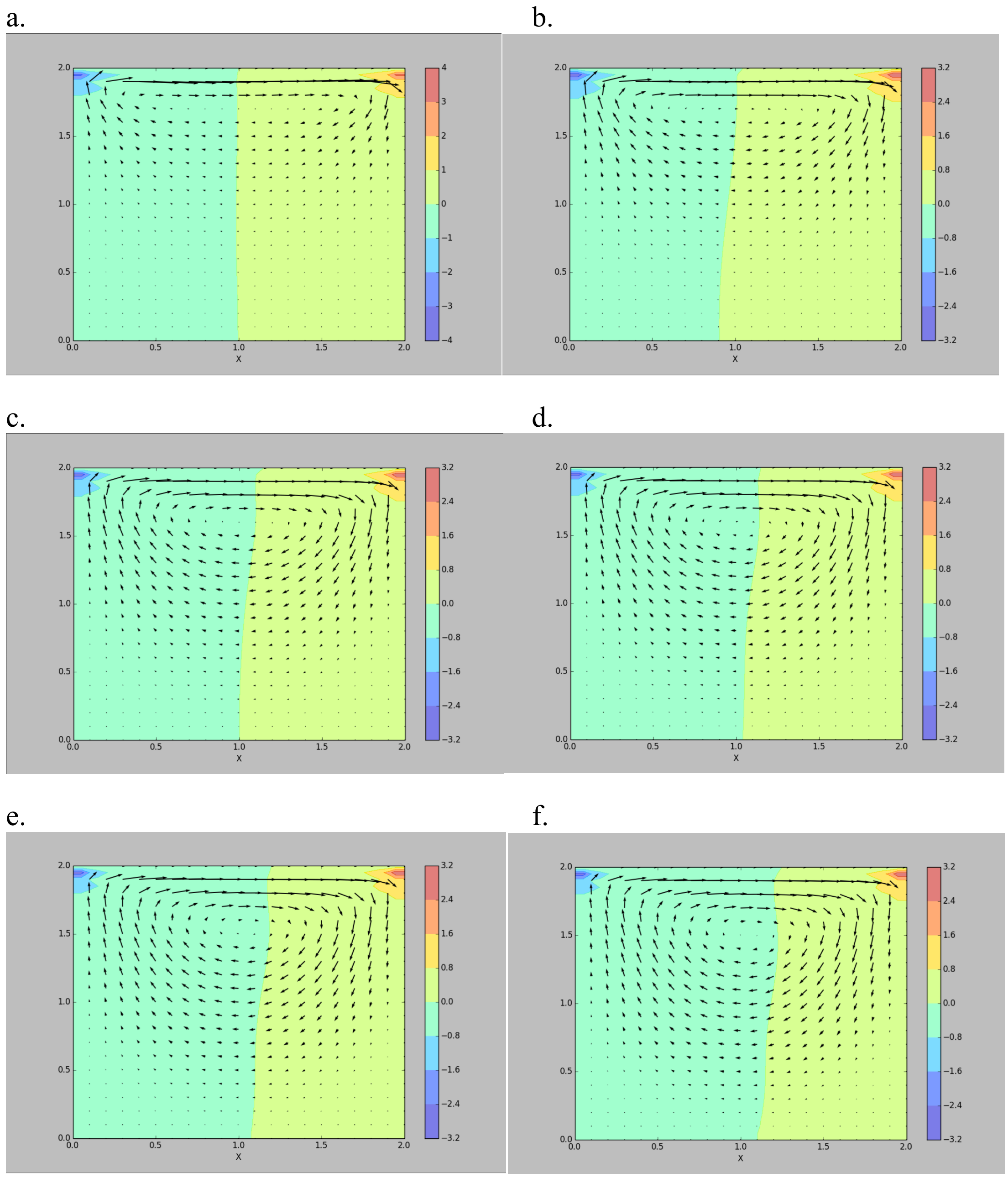

Figure 5: Cavity flow in different time steps (a. nt=100, b. nt=200, c. nt = 500 d. nt = 700, e. nt = 1000, f. nt = 1500)

Here two different pressure zones are available in both sides. For boundary condition, the images show that velocity decreases from the top to the bottom of the image. Here in figure 5(a), it has shown that the velocity is less and the circulation current is not so high. It increases with the increase of the time steps. In figure 5(f) it has shown that the velocity is very high with a pure circulation of flow.

Main objective of running this simulation is to prepare the web platform for other complicated simulation. The result can be summarized by the following way:

a. Django is well connected with other numerical and visualization library.
b. It can run the simulation in the server and can generate output for the simulation.
c. It can grab the data from the UI and by defining different parameter to different variable by GET method.
d. It can put different parameter in the CFD solver to get the output result.
e. Generated matplotlib image can be shown through Django in the browser.

6.2. Simulation for sedimentation tank

Experiment-1

The sedimentation tank consists of two phases. One is sewage and the other is the particle. So the experiment has been set for the multiphase flow simulation.
In this experiment, a slope has been made in the sedimentation tank. The depth of the inlet part has been modified and increased to 4m rather than 3.6m. The outlet part of the sedimentation tank has kept 3.6m as like the actual sedimentation tank. One baffle has been introduced in the sedimentation tank, which is 4m far from the inlet. The baffle has kept 2m from the top of the sedimentation tank. A three dimensional, steady state, implicit, pressure based solver has defined for this simulation. For the turbulence model standard $k - \varepsilon$ turbulence model was selected. Number of the phases was defined two. Primary phase has defined as sewage and the secondary phase has defined as sediment. The density and the viscosity of the sewage have set to 1000kg/m^3 and 0.0001kg/m·s respectively. On the other hand the density and the viscosity of the sediment have set to 1200kg/m^3 and 0.001 kg/m·s. Inlet velocity has set to 0.5m/s and the volume fraction has set to 0.05. After running simulation the volume fraction of the sediment has found as follows:

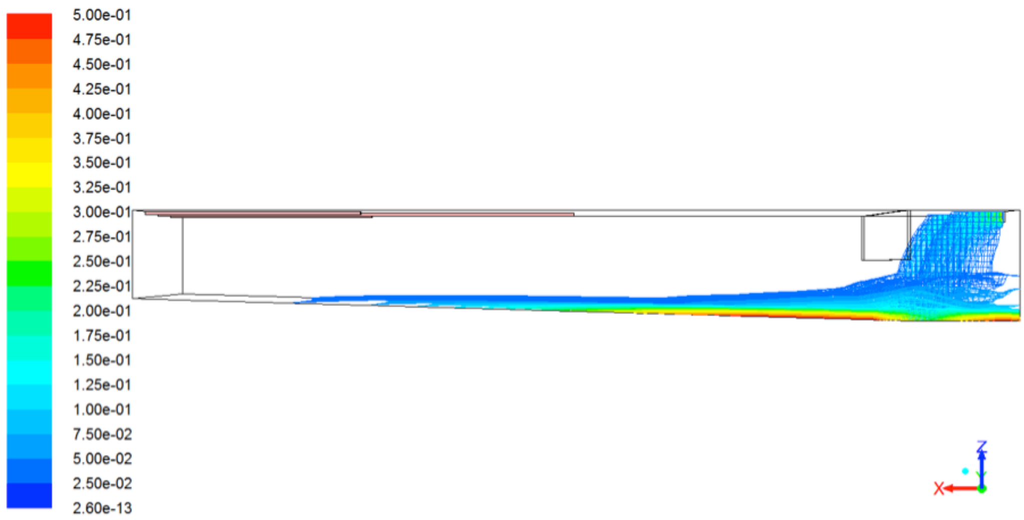

Figure 6: Volume fraction of the sediment after introducing slope and baffle.

This figure indicates that the volume of the sediment did not cover the total area of the sedimentation tank. Another significant result is the wastewater coming from the inlet has flashed away the sediment down the baffle. So the volume fraction of the sediment is lower in that zone. There is also a possibility of dead zone in that area. But for the baffle the dead zone didn't capture much area. If the sedimentation can be shown from the bottom the volume fraction would be clearer than before. One figure has taken from the bottom of the sedimentation tank has given below:

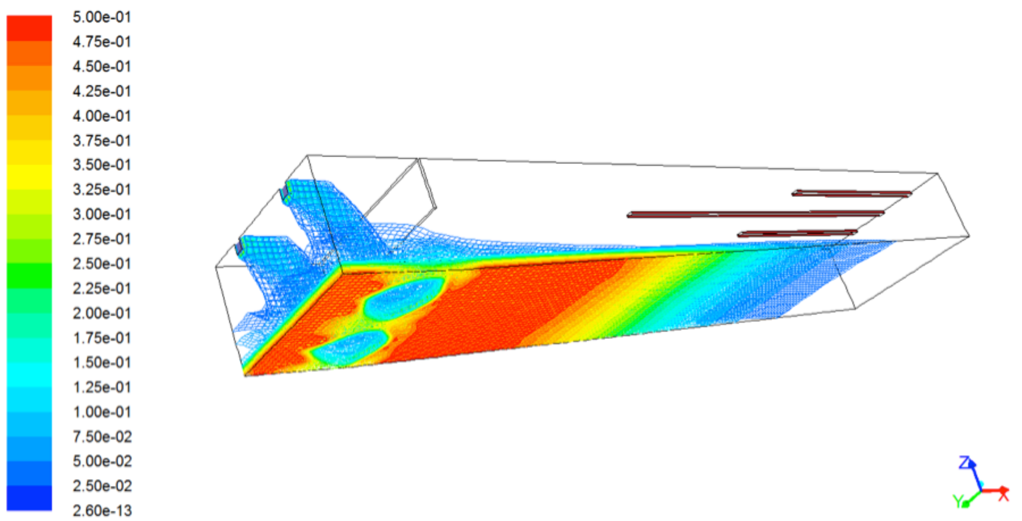

Figure 7: Volume fraction of the sediment from the bottom area.

This figure indicates more clearly about the volume of the sediment. Here it has seen that the water coming from the inlet washed away and reduce the volume of the sediment. In that way, introducing baffle can increase the efficiency of the sedimentation tank.

Experiment - 2

In this experiment two baffles have been introduced in the sedimentation tank. One baffle has set to 4m far from the inlet and it has set from the upper side. Another baffle has set 8m far from the inlet. The main objective of using this two baffle is to reduce the velocity of the sewage, which could be efficient for particle settlement after the baffle. Parameter has set same like experiment one for the simulation. After running the simulation the result found has discussed below:

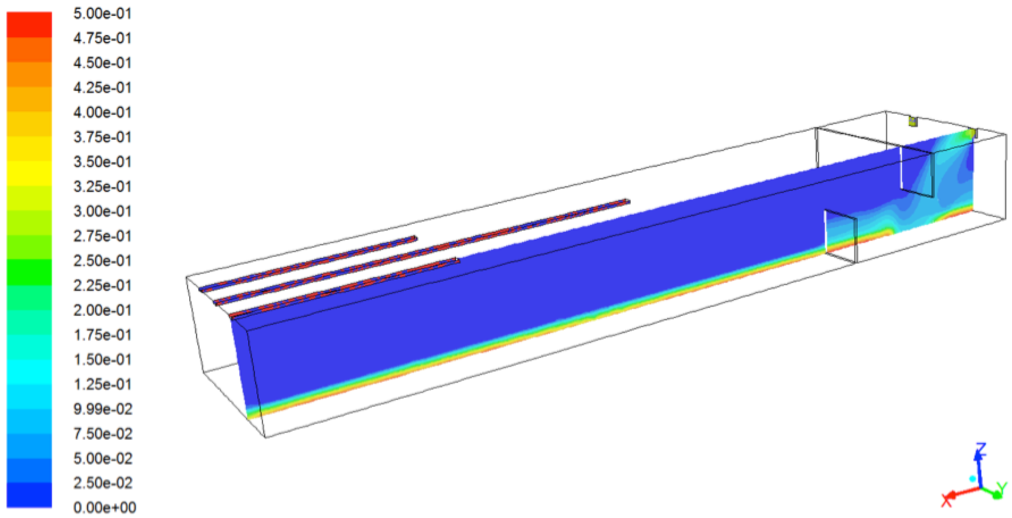

Figure 8: Cross section of the volume fraction of sediment using 2 baffles.

Figure 8 indicates that the volume of the sediment in the sewage body have reduced after second baffle. So the rest of the particles, those pass through the second baffle can be settled more efficiently.

Different parameter has been used for CFD simulation in sedimentation tank. For developing CFD web application these parameters need to be included with the UI of web application.

6.3. Architecture of the web application for sedimentation tank

Web interface of the application has kept as simple as possible. All the complexities have kept behind the scene. So the user will experience a clean and understandable UI. When users redirected to the web application, they need to register or login to their own workflow. Here user will see the previous works done by them and an option to open new project. By pressing 'new project' they will open a new project window where user will put necessary initial condition and boundary condition for the sedimentation tank. User also put the simulation data in the user interface. After pressing run simulation, user will wait for the simulation result.

User interface – 1

This is the first page of the web application. When user put the URL in the web browser they will redirect to this page. User will login to reach their own workflow. New user will register to reach their workflow.

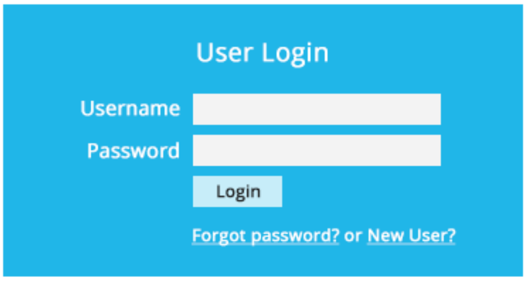

Figure 9: Login page in the user interface

User interface – 2

The user will see this page after logging in to the web application. Here user will see the works they have done before. They can see the previous project details by clicking specific project. By clicking the new project they will redirect to the new project interface.

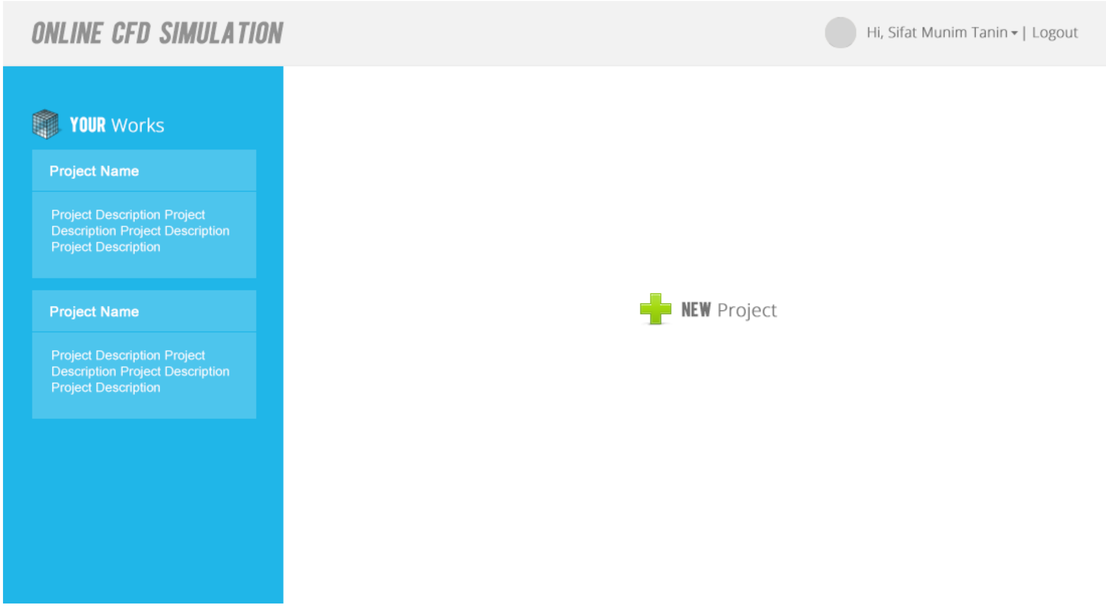

Figure 10: Main workflow page of the user

User interface – 3

New project user interface have shown here. Here user will put the simulation data and the mesh data for the simulation.

Mesh selection

The mesh selection part has divided into two parts. One part is for rectangular mesh and other part is non-rectangular mesh. In the rectangular mesh, user can change the height, width and depth. These changes will be done by accessing existing mesh file in the mesh library. User can also choose non-rectangular mesh. Here different option could come about the shape of the mesh and user can change the shape of the mesh by changing those options.

Simulation data

In the simulation data part, user will include necessary data for running the simulation. This data will be saved in the database. When OpenFOAM get the newly generated or existing mesh file, which is specified for the simulation, it will grab the data from the database and run the simulation. In the simulation data, many simplification have made in comparison to Fluent desktop application. User only can run simulation for two phases. This simulation will be solid-liquid simulation. The primary phase will be the sewage and the secondary phase will be the solid particle. Solver for running simulation will be three dimensional, transient, implicit, pressure-based solver. It will help the user to run the simulation in different time-steps. User will also need to define the density and the viscosity of the sewage and sediment. In addition, user will provide other data for inlet velocity, volume fraction of the sediment, time-step size, number of time-step and iteration per time-step.

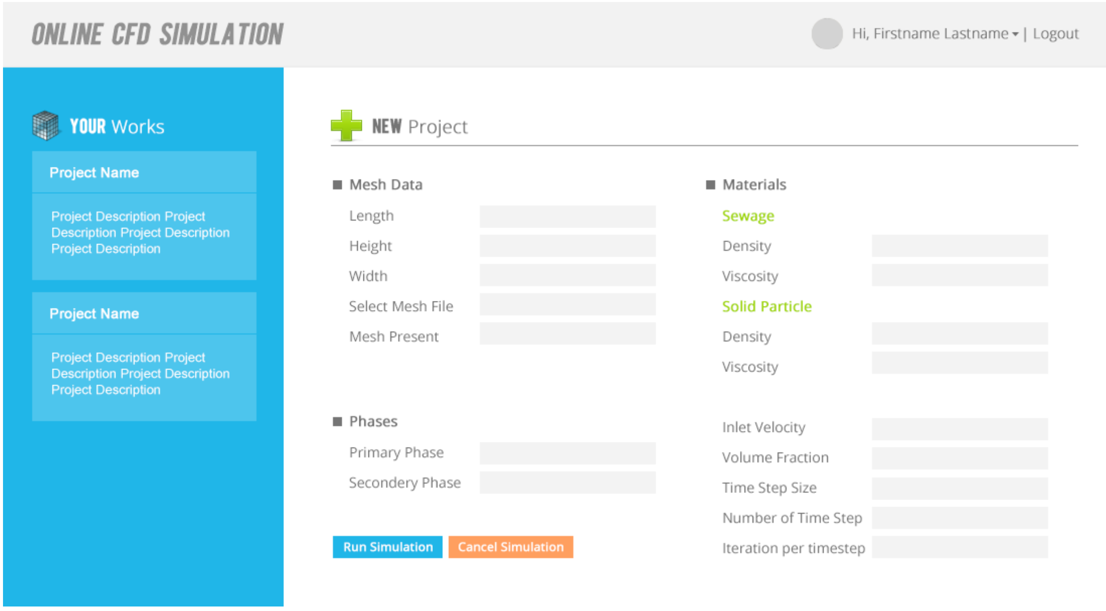

Figure 11: Data input page for new project

Output image

After defining all the data for simulation, user will press the 'run simulation' option. User will see a busy bar in his workflow in that case. Now user can press the back button and come to the main workflow page. User will also get option to cancel the project. By pressing cancel button, user can cancel the simulation process. During simulation process user can log out from the user interface although the simulation will be continuing. When the simulation will be finished, user will see the output image. User will click the project from 'your works' part and output image will be shown as a result(ParaViewWeb architecture www...).

The architecture of the web application is given below:

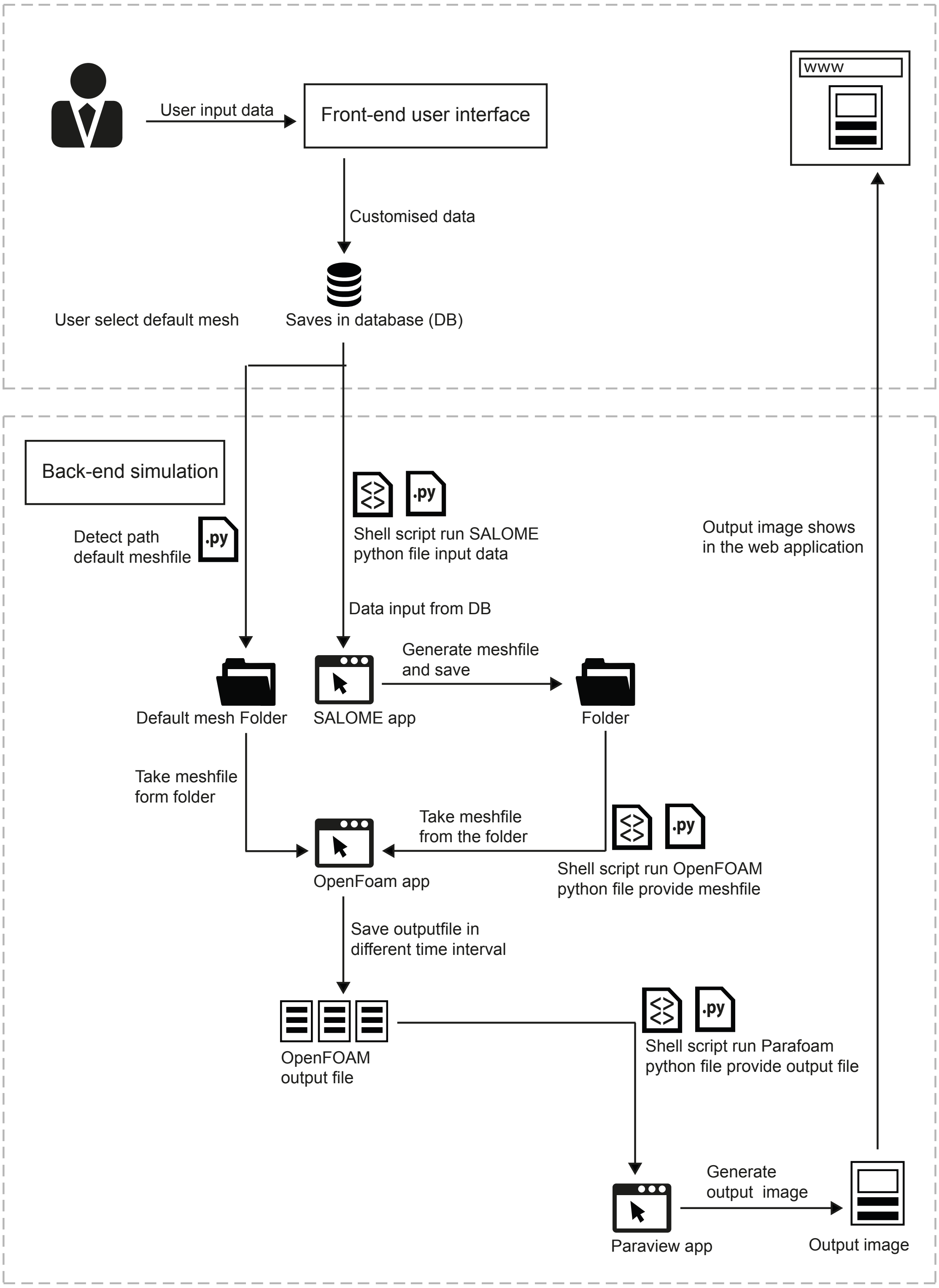

Figure 12: Structure of the web application for CFD applications

Chapter - 8
Discussion

Limitations and future possibility

Web application has become very popular in last few decades. Now a days cloud based service has developed with their high processing capability and helps to complete many complicated task through using web application. CFD simulation is one of them. It normally runs through desktop platform. There are very few CFD platforms available in the web. In this study one simple simulation have been run from Django web framework, which is cavity flow by neiver-stokes equation. For the simplicity, the simulation has not taken much time. Matplotlib, which is a 2d visualization library has been used to visualize the velocity and the pressure. Other visualization platform can be used for visualization. For example, mayavi3d could be used for visualizing the output. It has a 3d visualization capability and can be run by python programming language (Mayavi: 3D www…).

However, for the complex simulation like multiphase flow in the sedimentation, another CFD application has been planned to be integrated with this platform. Django web platform could also run this simulation but it needs proper CFD solver to run simulation. Discretizing complex CFD simulation model for numerical analysis could have a risk of accuracy and competency. In this case using a complete simulation package would be safe. Here those applications get priority, which can be accessed by python programming language. Mesh generation software SALOME can be integrated with OpenFOAM by pythonflu. Pythonflu act as a python wrapper for OpenFOAM. Another application was planned to be used to visualize the output image, which is called paraview. Paraview application is normally has used with OpenFOAM and it can be controlled by python script (ParaView and www…). It also has features for client-server execution through pvpython. Another important extension of paraview is ParaViewWeb, which is a API of paraview. By using this API(Application Programming Interface) web application can be embedded with 3D visualizing component. So it is possible to execute the dynamic 3d image form the web interface by using ParaViewWeb. In addition, ParaViewWeb need an apache server as a front-end. By setting up Django in apache server, this problem can be solved.

The application structure was planned to be implemented in the web interface, that might have some limitations in mesh generation. But this limitation could be removed by accessing mesh file properly. If every data of the mesh file is accessed properly, user could change many parameter of the mesh file from the user interface. In this case another mesh file will be generated from the parent mesh file for simulation.

On the other hand, a complex simulation process takes time to be completed. Here, one web platform need to control the simulation task step by step. In this case server computer have to be very powerful to analyze the data quickly. However, in a desktop CFD application one does not need to be concerned about the time. User can run a simulation and wait for the output. CFD desktop applications also give access to the users to analyze and store all kinds of data in their own way. Web application can limit this opportunity by simplifying the workflow. For example, in this study web interface has been designed to be implemented in a way that simplifies the simulation process in the sedimentation tank by creating a mesh library. Because from the user interface, user cannot access the mesh generation software directly. So the flexibility of the user is limited. If the mesh file can be accessed properly, user could get more flexibility for mesh generation.

Web platform also faces different security issues. There is also a chance of losing data in case of a major complication. Normally, CFD applications handle big sized file and images. So, low network system could be a problem with CFD web application. To analyze complex data, CFD desktop application is developed for powerful computer. But many users cannot have the opportunity to use such a computer. In case of web application the user does not need to think about the quality of the computer because the total process is completed in the cloud and the user can only see the output through browser. But all browser may not support this type of web application. Older browsers may cause problem to exibit properly. Browser compatibility could become a major issue in this case. Developer should test the application in a way that all browsers can support this application. Now-a-days many problems have been solved about browser compatibility. Most of the browsers have been kept in almost the same efficiency. If the users do not know the workflow of the application properly, they could face error in the application. In this case application should be tested in different method to solve all possible errors.

Different applications have been used to simulate the sedimentation tank in WWTP. CFD is also used in other unit to optimize it. But sedimentation tank has got priority because of its availability in most of the wastewater treatment plant. The efficiency of the sedimentation depends on different parameters. The geometry of the sedimentation also influences its efficiency. In this study, two multiphase flow simulations have been done by changing the geometry of the sedimentation tank. A baffle has been introduced in the sedimentation tank, which keeps most volume fraction of the sediment close to the inlet zone and increase the efficiency of the sedimentation tank. It can also reduce the velocity of the fluid.

To understand the physical behavior inside the sedimentation tank, computational fluid dynamics could be one of the important tools. However it could fail to provide actual result for assuming several parameters. For example, in this study multiphase flow simulation has been done for the sedimentation tank. Here the inlet velocity is kept constant for simplicity. However, in the real sedimentation tank inlet velocity increases and decreases with time. But it can provide some suggestions to the sedimentation tank designer. In this case, many CFD simulation need to be done by changing different parameter. It can lead to a possible conclusion for the designer.

Different physical problems like dead zone, eddy, short-circuiting can be found in the sedimentation. Running a simulation by understanding the actual situation can help in detecting the problem. Introducing baffle in a proper position can reduce the dead zone and increase the sedimentation area which intern increases the efficiency of the sedimentation tank.

Chapter - 8
Conclusion

The objective of the study is to find the possibility of developing a web interface for the WWTP experts, which can run the simulation for sedimentation tank. User interface of the web application needs to be easily understandable by which all the user does not face any problem with different terms. To fulfill this goal python programming language has been selected for web development. Python is a scripting language and has widely been used for scientific application development because it has a strong library for numerical analysis and calculation. However python needs another tool like CGI or WSGI to display the output in the webpage. To solve this problem, a web development framework called Django has been used. It supports WSGI, which helps the server understand the code and execute it. Python numerical library called NumPy have been used to generate mesh. Another library matplotlib has been used to plot different parameters. Here total process has been completed within the Django framework.

However, to ensure the accuracy of the numerical analysis and to visualize the output more elaborately, three open-source CFD applications have been selected to run the back-end part of the web application for sedimentation tank. They are SALOME, OpenFOAM and Paraview. Fluid flow in a sedimentation tank is a multiphase flow. Generating mesh is a vital part of the CFD simulation. One architecture of the web application has been designed by which users can create a mesh file form the user interface. Initial and the boundary condition will be declared in user interface.

Few simulations have been run to understand different parameters and flow behavior in the sedimentation tank. To run those simulation mesh file was needed to generate using mesh generating software. But in the web interface it is possible to skip or simplify this part by using mesh library. To fulfill this goal, it was planned to create possible mesh files for the mesh library. User will get the option to choose a mesh file in the user interface. When user chooses one mesh file from the user interface and run simulation, OpenFOAM can gather that mesh file from the server pc to complete the simulation. It has been planned to use Paraview to visualize the output image. Those images will be shown in the web interface as a result. So, users have only interaction with the web interface and the total simulation process will be completed in the back-end. Developing this type of web interface can make CFD easier to general user. Web interface could be developed for specific group of user to fulfill specific interest. This kind of web interface has the possibility of vast huge popularity for its simplicity. In addition, improvement of cloud-based service now ensures the better processing power which is capable of running complex CFD simulation from the web interface. In future, CFD will be appeared with more functionality and efficiency through cloud based service.

References

17.1. subprocess — Subprocess management(2014). Available at: http://docs.python.org/2/library/subprocess.html#popen-constructor (accessed: 23.03.2013)

Ahrens, J., Geveci, B. & Law, C. (2005). Paraview: An end-user tool for large data visualization. *The Visualization Handbook*, 717: 731.

Altair Announces Innovative Cloud Applications on Amazon Web Services for Mainstream CFD Users. (2013). Available at: http://www.altair.com/newsdetail.aspx?news_id=10865&news_country=en-US (accessed:20.02.2013).

Barba, L.A.(2013). *Step-2: Non-linear Convection*. Available at: http://nbviewer.ipython.org/github/barbagroup/CFDPython/blob/master/lessons/02_Step_2.ipynb (accessed: 23.03.2013)

Barrett, P., Hunter, J., Miller, J. T., Hsu, J.-C. & Greenfield, P. (2005). *matplotlib--A Portable Python Plotting Package*. Astronomical Data Analysis Software and Systems XIV. 91 pp.

Bergeaud, V. & Tajchman, M. (2007). *Application of the SALOME software architecture to nuclear reactor research*. Proceedings of the 2007 spring simulation multiconference-Volume 2: Society for Computer Simulation International. 383-387 pp.

Calculators to solve specific Computational Fluid Dynamics (CFD) problems. (2013). Available at: https://www.cfdcalc.com/html/calculators.html (accessed:20.02.2013).

CFDCalc Simple Pipe Calculator (2013). Available at: https://www.cfdcalc.com/doc/CFDCalc_Calculator_SimplePipe_Tutorial.pdf (accessed: 20.02.2013).

Chudoba, R., Sadílek, V., Rypl, R. & Vořechovský, M. (2013). Using Python for scientific computing: Efficient and flexible evaluation of the statistical characteristics of functions with multivariate random inputs. *Computer Physics Communications*, 184 (2): 414-427.

Dacolt Web Apps: Online combustion software (2004-2014). Available at: http://www.dacolt.com/webapps (accessed: 20.02.2013)

De Clercq, J., Jacobs, F., Kinnear, D. J., Nopens, I., Dierckx, R. A., Defrancq, J. & Vanrolleghem, P. A. (2005). Detailed spatio-temporal solids concentration profiling during batch settling of activated sludge using a radiotracer. *Water research*, 39 (10): 2125-2135.

Fan, L., Xu, N., Ke, X. & Shi, H. (2007). Numerical simulation of secondary sedimentation tank for urban wastewater. *Journal of the Chinese Institute of Chemical Engineers*, 38 (5): 425-433.

Features of OpenFOAM (2011-2014). Available at: http://www.openfoam.org/features/ (accessed: 20.02.2013).

Ghawi, A. H. & Kriš, J. A Computational Fluid Dynamics Model of Flow and Settling in Sedimentation Tanks.

Giacaman, N. & Sinnen, O. (2013). Multiprocessing with GUI-awareness using OpenMP-like directives in Java. *Parallel Computing*.

Goula, A. M., Kostoglou, M., Karapantsios, T. D. & Zouboulis, A. I. (2008). The effect of influent temperature variations in a sedimentation tank for potable

water treatment—a computational fluid dynamics study. *Water research*, 42 (13): 3405-3414.
Heat Sink Calculator with 3D animation (2013). Available at: https://www.cfdcalc.com/html/heatsink.html (accessed: 20.02.2013).
Holovaty, A. & Kaplan-Moss, J. *The Django Book, 2008. Recurs electrònic*.
Holovaty, A. & Kaplan-Moss, J. (2009). The Definitive Guide to Django. *Estados Unidos: Editorial Apress*: 72-73.
HOWTO Use Python in the web (2014). Available at: http://docs.python.org/2/howto/webservers.html (accessed: 23.03.2013)
Hvitved-Jacobson, T. & Yousef, Y. (1991). Highway runoff quality, environmental impacts and control. *Highway pollution*, 44.
Imam, E., McCorquodale, J. A. & Bewtra, J. K. (1983). Numerical modeling of sedimentation tanks. *Journal of hydraulic engineering*, 109 (12): 1740-1754.
Jasak, H., Jemcov, A. & Tukovic, Z. (2007). *OpenFOAM: A C++ library for complex physics simulations*. International workshop on coupled methods in numerical dynamics. 1-20 pp.
Kahane, R., Nguyen, T. & Schwarz, M. (2002). CFD modelling of thickeners at Worsley Alumina Pty Ltd. *Applied Mathematical Modelling*, 26 (2): 281-296.
Kawamura, S. (1991). *Integrated design of water treatment facilities*: Wiley New York.
Landau, L. (1987). *Fluid mechanics: volume 6 (course of theoretical physics) Author: LD Landau, EM Lifshitz, Publisher: Bu*: Butterworth-Heinemann.
Langtangen, H. P. (2006). *Python scripting for computational science*, vol. 3: Springer.
Maus, C. & Uhl, M. (2010). *Tracer Studies for the Modelling of Sedimentation Tanks*, Muenster University of Applied Sciences, Laboratory of Water Resources Management, Germany.
Mayavi: 3D scientific data visualization and plotting in Python. Available at: http://docs.enthought.com/mayavi/mayavi/ (accessed: 12.03.2014)
Mesh Generation. Available at: http://www.openfoam.org/features/mesh-generation.php (accessed: 23.03.2013)
New cloud-based CFD simulation toolkit (2013). Available at: http://www.engineering.com/DesignSoftware/DesignSoftwareArticles/ArticleID/6341/New-Cloud-Based-CFD-Simulation-Toolkit.aspx (accessed: 20.03.2013)
Oliphant, T. E. (2006). *A Guide to NumPy*, vol. 1: Trelgol Publishing USA.
ParaViewWeb architecture. Available at: http://www.paraview.org/ParaView3/Doc/Nightly/www/js-doc/index.html#!/guide/architecture (accessed:12.03.2014)
ParaView and Python. Available at: http://www.kitware.com/media/html/ParaViewAndPython.html (accessed:12.03.2014)Plekhanova, J. (2009). "Evaluating web development frameworks: Django, ruby on rails and cakephp. *Institute for Business and Information Technology*.
Population (Total).(2014). Available at: http://data.worldbank.org/indicator/SP.POP.TOTL/countries?display=graph (accessed:20.02.2013).
Programming Language Popularity (2013). Available at: http://langpop.com/ (accessed: 20.03.2013)

Pythonflu (2010) Available at: http://pythonflu.wikidot.com/ (accessed: 20.02.2013)
Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12 (3): 23-49.
Razmi, A., Firoozabadi, B. & Ahmadi, G. (2009). Experimental and Numerical Approach to Enlargement of Performance of Primary Settling Tanks. *Journal of Applied Fluid Mechanics*, 2 (1).
Righetti, M. & Romano, G. (2004). Particle–fluid interactions in a plane near-wall turbulent flow. *Journal of Fluid Mechanics*, 505: 93-121.
Rodríguez López, P., Lavín, A. G., Mahamud López, M. M. & Bueno de las Heras, J. L. (2008). Flow models for rectangular sedimentation tanks. *Chemical Engineering and Processing: Process Intensification*, 47 (9): 1705-1716.
Schamber, D. R. & Larock, B. E. (1981). Numerical analysis of flow in sedimentation basins. *Journal of the Hydraulics Division*, 107 (5): 575-591.
Sempolinski, P., Thain, D., Wei, D. & Kareem, A. (2012). *A system for management of Computational Fluid Dynamics simulations for civil engineering*. E-Science (e-Science), 2012 IEEE 8th International Conference on: IEEE. 1-8 pp.
Shahrokhi, M., Rostami, F., Md Said, M. A. & Sabbagh Yazdi, S. R. (2012). The effect of number of baffles on the improvement efficiency of primary sedimentation tanks. *Applied Mathematical Modelling*, 36 (8): 3725-3735.
Shahrokhi, M., Rostami, F. & Md Said, M. A. (2013). Numerical modeling of baffle location effects on the flow pattern of primary sedimentation tanks. *Applied Mathematical Modelling*, 37 (6): 4486-4496.
SMX Mixer Calculator with 3D Flow Simulation (2013). Available at: https://www.cfdcalc.com/html/smxmixer.html (accessed: 20.02.2013).
Stamou, A., Adams, E. & Rodi, W. (1989). Numerical modeling of flow and settling in primary rectangular clarifiers. *Journal of hydraulic research*, 27 (5): 665-682.
Standard Solvers. Available at: http://www.openfoam.org/features/standard-solvers.php (accessed: 23.03.2013)
Stamou, A. I. (2008). Improving the hydraulic efficiency of water process tanks using CFD models. *Chemical Engineering and Processing: Process Intensification*, 47 (8): 1179-1189.
Tamayol, A., Firoozabadi, B. & Ahmadi, G. (2008). Effects of inlet position and baffle configuration on hydraulic performance of primary settling tanks. *Journal of Hydraulic Engineering*, 134 (7): 1004-1009.
Tarpagkou, R. & Pantokratoras, A. (2013). CFD methodology for sedimentation tanks: The effect of secondary phase on fluid phase using DPM coupled calculations. *Applied Mathematical Modelling*, 37 (5): 3478-3494.
Tu, J., Yeoh, G. H. & Liu, C. (2007). *Computational fluid dynamics: a practical approach*: Butterworth-Heinemann.
von Krogh, G. & Spaeth, S. (2007). The open source software phenomenon: Characteristics that promote research. *The Journal of Strategic Information Systems*, 16 (3): 236-253.
Wang, X., Yang, L., Sun, Y., Song, L., Zhang, M. & Cao, Y. (2008). Three-dimensional simulation on the water flow field and suspended solids concentration in the rectangular sedimentation tank. *Journal of Environmental Engineering*, 134 (11): 902-911.
Wills, R. & Davis, C. (1962). *Flow patterns in a rectangular sewage sedimentation tank*. 1st International Conference on Water Pollution research.