University of Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: Offshore Technology/ Subsea Technology	Spring semester, 2015 Open / ~~Restricted access~~
Writer: Jessica Rikanti Tawekal	…… (Writer's signature)
Faculty supervisor: Bjørn H. Hjertager External supervisor(s):	
Thesis title: CFD Simulation of the Flow over a 2-Dimensional Pipe and Vortex Induced Vibration of the Pipe with 1 Degree of Freedom	
Credits (ECTS): 30	
Key words: *vortex shedding, laminar flow, turbulent flow, seabed proximity, vortex induced vibration, Strouhal number, circular cylinder, computational fluid dynamic, OpenFOAM*	Pages: XII + 96 + enclosure: 28 + CD Stavanger, June 15th 2015

ABSTRACT

The fluid that flows through a pipeline free span can generate vortex shedding around the pipe and triggers Vortex Induced Vibration (VIV). This VIV is a major source of fatigue damage due to dynamic stresses in the pipeline free span. Therefore, it is necessary to observe the vortex shedding around the pipe since it could help to understand the VIV better.

The objective of this thesis is to simulate a steady uniform flow that flows over a 2-dimensional pipe, which in this case is represented by a circular cylinder, with different parameters such as Reynolds number and span height using a computational fluid dynamic software called as OpenFOAM. Flow with different Reynolds numbers are simulated as laminar flow ($20 \leq Re \leq 1000$) and turbulent flow ($10^4 \leq Re \leq 10^7$), and then it will be modelled to flow over a free fixed pipe. Afterwards, a fixed pipe with seabed proximities will be simulated in a laminar flow ($Re = 400$). Lastly, VIV simulations will be done for a free 2-dimensional pipe with 1 degree of freedom enabled in a laminar flow ($Re = 500$). In the VIV simulation, the mass and stiffness of the pipe will be modified to see how it will affect the VIV behaviors.

For flow over a free fixed pipe in laminar flow, the value of drag coefficients stays almost always constant and the amplitude of the lift coefficients increases as the Reynolds number increases. The Strouhal – Reynolds number relationships of the laminar flow simulation have a good agreement with other experimental results.

The Strouhal – Reynolds number relationships of the turbulent simulation have a good agreement with the Strouhal – Reynolds number relationships from other literature. However, the drag coefficients are much higher than the drag coefficients for cylinders that were given by Cengel and Cimbala (2010). Conclusively, the turbulent flow simulation still need improvements, one of it is by changing the turbulence model from RANS to LES that can handle sophisticated vertices.

The span height of the pipeline free span suppresses the vortex shedding phenomenon as the gap ratio gets smaller. The drag coefficients decreases as the gap ratio gets smaller, while the lift coefficient has a unique relationship with the gap ratio, as described by Sumer and Fredsøe (1997).

However, the simulation results in this thesis could not draw the same relationship due to lack of simulation data.

In the VIV simulations, when the reduced velocity reaches 4, the pipe is in resonance. Conclusively, we shall keep the reduced velocity to be always less than 4 to avoid this resonance. This can be done by modifying the Eigen frequency, which is affected by the mass and stiffness of the pipe.

Keywords: *vortex shedding, laminar flow, turbulent flow, seabed proximity, vortex induced vibration, Strouhal number, circular cylinder, computational fluid dynamic, OpenFOAM*

Acknowledgements

This thesis is submitted as partial fulfillment for the requirements of Master of Science (M.Sc.) at the University of Stavanger (UiS), Norway. It would not have been possible to complete this thesis without assistance, support, guidance, and help from many individuals. Therefore, I would like to express my gratitude towards all of them.

I would like to thank Mr. Bjørn H. Hjertager for accepting my thesis proposal even though he knew about my limited background and knowledge about CFD. Mr. Bjørn had patiently guided me and gave me a lot of advices whenever I am lost or stuck with the simulation. Indeed, without his support and guidance during my thesis writing, I will not be able to complete it on time and graduate like this.

I would also like to thank Mr. Knut Erik Giljarhus for always guiding me with the OpenFOAM and always tries his best to answer all of my questions during my thesis writing.

I want to thank my family as well, especially my parents, for always guiding, supporting, and their prayers during my thesis writing. They are always worried about me to the extent that they are willing to spare their time and money to come here merely for seeing how I am doing in Stavanger. Even as I am writing this acknowledgement, they are sleeping soundly in my room, tired due to hours of travel and jet lag; all of these just so that they could come to my graduation.

Last but not least, thank you to my fellow Indonesians in Stavanger (Ade, Beki, Tina, Andika, Faras, Gilang, and Hafiz) for the joy that we have had during our stay in Stavanger. I am truly grateful that I came to Stavanger with them, because I know that my stay would not have been as great as it is now if it were other people. I have learned a lot of things (especially cooking ☺) from them and I definitely will miss our time together.

Stavanger, June 2015

Jessica Rikanti Tawekal

Table of Contents

Nomenclature

General

- Symbols are generally defined right after they appear in the text
- Only the most used symbols are listed in the following section
- Over-dots signify differentiation with respect to time

Alphabet

A_R	Reference area
c	Linear viscous damping
C_a	Added mass coefficient
C_D	Drag coefficient
$\bar{C}_D$	Mean drag coefficient
C_L	Lift coefficient
D	Outer diameter of the pipe
e	Gap between the seabed and pipe span
e/D	Gap ratio
f_n	Eigen frequency
f_{osc}	Oscillation frequency
f_v	Vortex shedding frequency
F_D	Drag forces
$\bar{F}_D$	Mean drag forces
F_L	Lift forces
H	The height of the domain
i	Internal energy
k	Stiffness
L	The horizontal length of pipe
L_d	The downstream length of the domain
L_u	The upstream length of the domain
m	Mass
m_a	Added mass
m_{dry}	Weigh of pipe in the air
p	Pressure
Re	Reynolds number

St	Strouhal number
S_m	Source/sink term
t	Time
T_n	Natural period
T_v	Vortex shedding period
u	Velocity in x-direction
$\vec{u}$	Velocity vector
U_∞	Free stream velocity
U	Velocity
v	Velocity in y-direction
V_R	Reduced velocity
w	Velocity in z-direction
y	Vertical displacement
y/D	Non-dimensional vertical displacement

Symbols

β	Frequency ratio
δ	Rate of linear deformation
υ	Kinematic viscosity
ρ	Density
τ_{ij}	The stress component acts in the *j*-direction on the surface normal to the *i*-direction
μ	Dynamic viscosity, relates stresses to linear deformations
λ	Relative damping
ω_d	Damped angular frequency of oscillation
ω_n	Eigen angular frequency
ϕ	General property
τ	Viscous stress
θ	Phase angle
Γ	Diffusion coefficient

Abbreviation

DAF	Dynamic amplification factor
RANS	Reynolds-averaged Navier-Stoke
VIV	Vortex induced vibration

List of Figures

List of Tables

CHAPTER 1 - Introduction

1.1. Background and Motivation

The fluid that flows through a pipeline free span can generate sheet vortices in the wake and later on it will be shed alternately from top to bottom of the pipe, as seen in Figure 1.1. This phenomenon is commonly known as vortex shedding. Generally, the fluid that flows over a pipeline is a combination of the tidal current and wave induced velocity. However, the current tends to dominate in deep water due to the decaying wave induced velocities along with the increasing water depth.

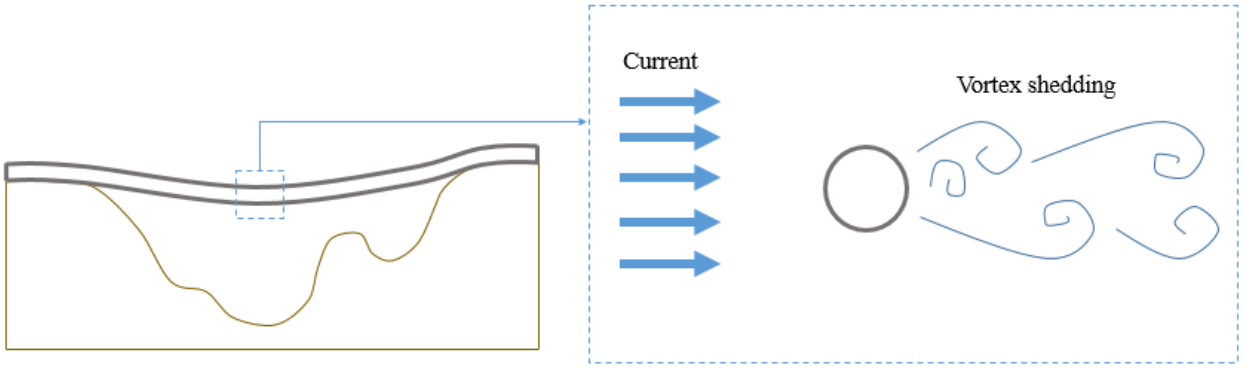


Figure 1.1: Vortex shedding phenomena behind the pipe span

The vortex shedding phenomenon may cause the span to vibrate and this event is commonly known as the Vortex Induced Vibration (VIV). VIV is a major source of fatigue damage due to the dynamic stresses that happen in the pipeline free span. Observing and understanding the vortex shedding around the pipe could help to understand the VIV better since it is the phenomenon that triggers the VIV on a pipe.

In the real field, the flow characteristics will depend on the condition of the tidal current and the wave, which are often varied due to the seasonal change. Hence the vortex shedding observation will be done for various flow characteristics, which in this thesis is represented by the Reynolds numbers.

The span height of pipeline free span will also vary due to the seabed unevenness and therefore, it will also be interesting to see the effect of the span height to the vortex shedding phenomenon around the pipe. Additionally, it is also interesting to observe the VIV behavior of the pipe when the mass and stiffness of the pipe are modified.

Observing the vortex shedding around the pipe could be done by either doing experiments or simulations. Since doing experiments is time consuming and will cost a lot of money, simulations are definitely more advantageous for this thesis. Computation Fluid Dynamics (CFD), which is a numerical modelling in fluid dynamics, will be used in the analysis since it can produce an extremely large volume of results at virtually no added expense (Versteeg & Malalasekera, 2007). A toolbox called OpenFOAM (Open Field Operation and Manipulation) is an example of free, open source CFD software package which has a large user base across most areas of engineering and science, from both commercial and academic organizations.

1.2. Scope

The simulations in this thesis will be divided into 2 major parts:

1. Simulation of Flow Over a Fixed Pipe

- Laminar Flow

A steady current with various low Reynolds numbers ($Re \leq 1000$) will flow through a free 2-dimensional fixed pipe.

- Turbulent Flow

A steady current with various high Reynolds numbers ($10^4 \leq Re\ 10^7$) will flow through a free 2-dimensional fixed pipe.

- Pipe with seabed proximity

A steady current with a particular Reynolds number ($Re = 400$) will flow through a 2-dimensional pipe with various span heights.

2. Simulation of Vortex Induced Vibration of a Pipe

A steady current with a particular Reynolds number ($Re = 500$) will flow through a free 2-dimensional pipe and later on the pipe will oscillate due to the vortex shedding that is created

behind it. In this case, the pipe is set to only have one degree of freedom, which is a vertical translation motion.

1.3. Objectives

Generally, the purpose of this thesis is to create a CFD simulation of flow over a foxed pipe and VIV of a pipe. The objectives of flow over a fixed pipe simulation are as follows:

- Measure the vortex shedding frequency and force coefficients from each simulation
- Calculate the Strouhal number of each simulation
- Analyze the relationship between the force coefficients and the Reynolds number
- Analyze the relationship between the Strouhal number and the Reynolds number
- Analyze how the span height of the pipe would affect the vortex shedding, including the Strouhal number and the force coefficients

The objectives of VIV simulation are as follows:

- Analyze how different mass and stiffness will affect the vortex shedding around the pipe

1.4. Organization of Project Report

- Chapter 2: Presents the theoretical background regarding necessary parameters in vortex shedding phenomena and VIV.
- Chapter 3: Presents the theories regarding CFD and brief explanations about OpenFOAM and its solvers.
- Chapter 4: Presents the case setup for flow regimes simulation; laminar flow, turbulent flow, and pipe with seabed proximity
- Chapter 5: Presents the results and discussion of the flow regimes simulation
- Chapter 6: Presents the case setup for VIV case, the results of the simulation, and discussion
- Chapter 7: Presents the overall conclusion of the simulation and suggestion for future work.

CHAPTER 2 - Theoretical Background for Flow Over a Cylinder

2.1. No-slip Condition

When a fluid flows through a solid surface, it will come to a complete stop at the surface and the velocity relative to the surface (both normal and tangential) is assumed to be zero. This condition, where a fluid is in direct contact with a solid and "sticks" onto the surface, is commonly known as the no-slip condition (Cengel & Cimbala, 2010). The viscosity of the fluid greatly contributes in defining the property of the non-slip condition.

Because of the viscous force between the fluid layers, the fluid layer that "sticks" to the solid surface in a non-slip condition slows the adjacent fluid layer, and then slows the next layer, and so on (Cengel & Cimbala, 2010). Therefore, according to Cengel and Cimbala (2010), all velocity profiles of fluids must have zero values when it is in contact with a solid surface and the non-slip condition is responsible for this velocity profile's development (as shown in **Error! Reference source not found.**).

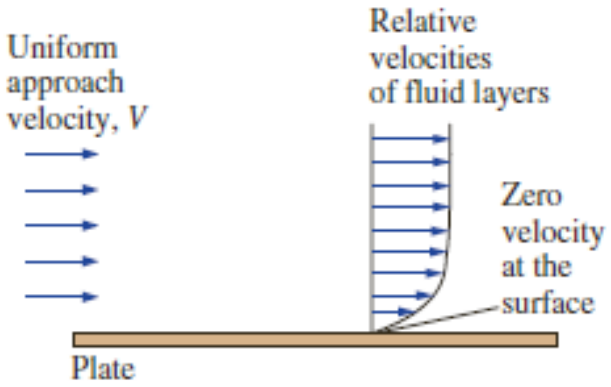


Figure 2.1: Fluid velocity profile during the non-slip condition (Cengel & Cimbala, 2010)

When a fluid flows over a curved surface (e.g. circular cylinder) the boundary layer might be no longer attached to the surface and it separates from the surface instead – this process is called as flow separation (as shown in Figure 2.2).

Figure 2.2: Flow separation during flow over a curved surface (Cengel & Cimbala, 2010)

2.2. Flow Regimes

The Reynolds number is a non dimensionless hydrodynamic number that is used to describe the flow around a smooth circular cylinder. According to Versteeg and Malalasekera (2007), the Reynolds number indicates a measure of the relative importance of inertia forces and viscous forces. Therefore, the Reynolds number, Re, is formulated as the ration of the inertia forces to viscous forces as follows

$$\mathrm{Re} = \frac{U \cdot y}{\upsilon} \tag{1}$$

Where y indicates the distance in the cross stream direction, υ indicates kinematic viscosity, and U indicates the free stream velocity. For flow over a pipe case, the distance in the cross stream direction is equal to the diameter of the pipe.

$$\mathrm{Re} = \frac{U \cdot D}{\upsilon} \tag{2}$$

Significant changes of the Reynolds number will result on different flow regimes. According to Versteeg and Malalasekera (2007), when the Reynolds number is below the so-called critical Reynolds number, a highly ordered fluid motion with smooth layers of fluid as its characteristic will appear. And the flow is also steady, which means that the applied boundary conditions do not change with the time. This kind of flow regime is called as laminar flow.

When the value of the Reynolds number is above the critical Reynolds number, a complicated series of events takes place which eventually leads to a radical change of the flow character (Versteeg & Malalasekera, 2007). In the final state, the fluid motion will become highly disordered and chaotic. Consequently, the velocity and all other flow properties will also vary in a random and chaotic way. This kind of flow regime is called as turbulent flow.

In addition, a flow that alternates between being laminar and turbulent is called as transitional flow (Cengel & Cimbala, 2010). The visualizations of laminar, transitional, and turbulent flow are presented in Figure 2.3.

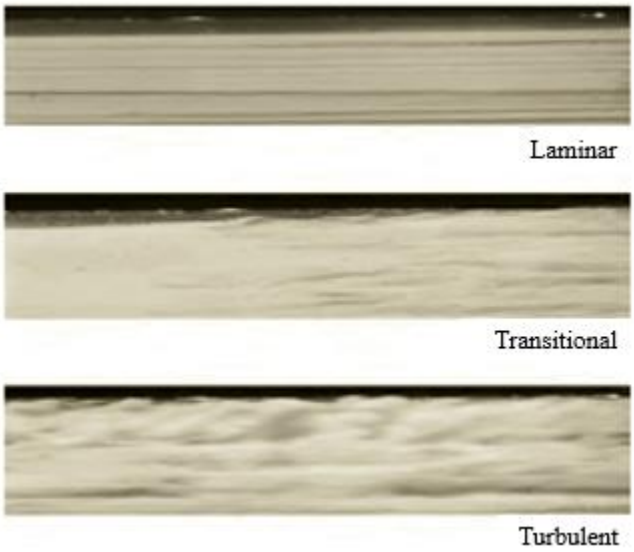
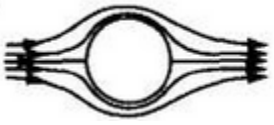
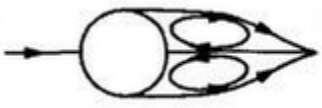
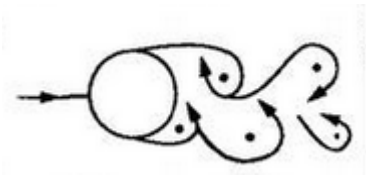
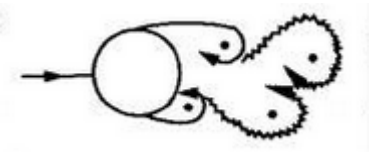
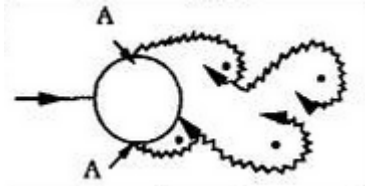
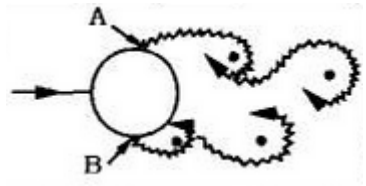
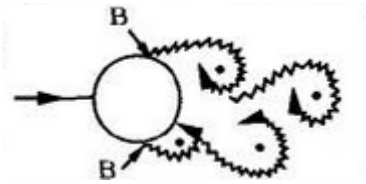
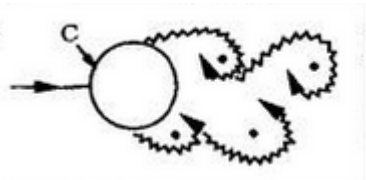
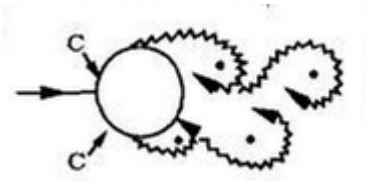


Figure 2.3: Laminar, transitional, and turbulent flows (Cengel & Cimbala, 2010)

According to Sumer and Fredsøe (1997), the changes of Reynolds number will create flow separation in the wake region of the cylinder, which is commonly known as vertices. At low Reynolds number (Re < 5), no separation occurs. However, if the Reynolds number is further increased, the separation will start to occur and become unstable, and later it will initiate a phenomenon called vortex shedding at a certain frequency. Consequently, the wake has an appearance of a vortex street as shown in Table 2.1

Table 2.1: Regimes of flow around a smooth circular cylinder in steady current (Sumer & Fredsøe, 1997)

No	Flow Regimes	Description	Range of Re
a)		No separation. Creeping Flow	$Re < 5$
b)		A fixed pair of symmetric vortices	$5 < Re < 40$

No	Flow Regimes	Description	Range of Re
c)		Laminar vortex street	$40 < Re < 200$
d)		Transition to turbulence in the wake	$200 < Re < 300$
e)		Wake completely turbulent. A: Laminar boundary layer separation	$300 < Re < 3 \cdot 10^5$ Subcritical
f)		A: Laminar boundary layer separation B: Turbulent boundary layer separation; but boundary layer laminar	$3 \cdot 10^5 < Re < 3.5 \cdot 10^5$ Critical (Lower Transition)
g)		B: Turbulent boundary layer separation; the boundary layer partly laminar partly turbulent	$3.5 \cdot 10^5 < Re < 1.5 \cdot 10^6$ Supercritical
h)		C: Boundary layer completely turbulent at one side	$1.5 \cdot 10^6 < Re < 4 \cdot 10^6$
i)		C: Boundary layer completely turbulent at two sides	$4 \cdot 10^6 < Re$ Transcritical

2.3. Vortex Shedding

According to Sumer and Fredsøe (1997), the vortex shedding phenomenon appears when pairs of stable vertices are exposed to small disturbances and become unstable. This phenomenon appears for values of Re greater than 40, where the boundary layer over the cylinder surface will separate due to the adverse pressure gradient imposed by the divergent geometry of the flow environment at the rear side of the cylinder (Sumer & Fredsøe, 1997). Consequently, a shear layer is formed and a boundary layer is also formed along the cylinder.

The boundary layer contains a significant amount of vorticity and it is fed into shear layer formed downstream of the separation point (refer to Figure 2.4a). According to Sumer and Fredsøe (1997), this event causes the shear layer to roll up into a vortex with a sign identical to that of the incoming vorticity (refer to Figure 2.4b).

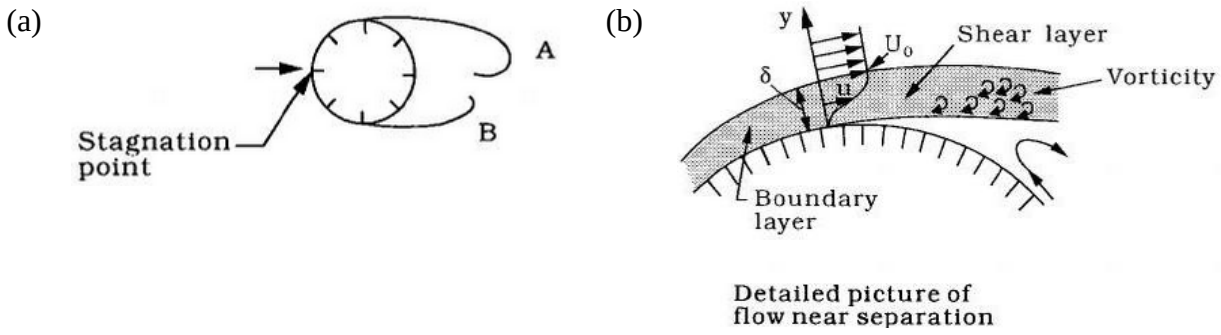


Figure 2.4: The shear layer of the flow over a cylinder (Sumer & Fredsøe, 1997)

It was mentioned previously that the vortex shedding occurs at a certain frequency, which is called as the vortex shedding frequency, f_v. This frequency can be seen as a function of the Reynolds number and is formulated as the inverse of the vortex shedding period, T_v.

$$f_v = 1/T_v \tag{3}$$

The normalized vortex-shedding frequency is called as Strouhal number, *St*, and is formulated as follows

$$St = \frac{f_v \cdot D}{U} \tag{4}$$

In words, Koushan (2009) stated that Strouhal number indicates the relation between the vortex shedding frequency (f_v) of a fixed cylinder/pipe and the free stream velocity divided by the

cylinder's diameter (U/D). Additionally, the relationship between Reynolds number and Strouhal number for a cylinder is presented in Figure 2.5.

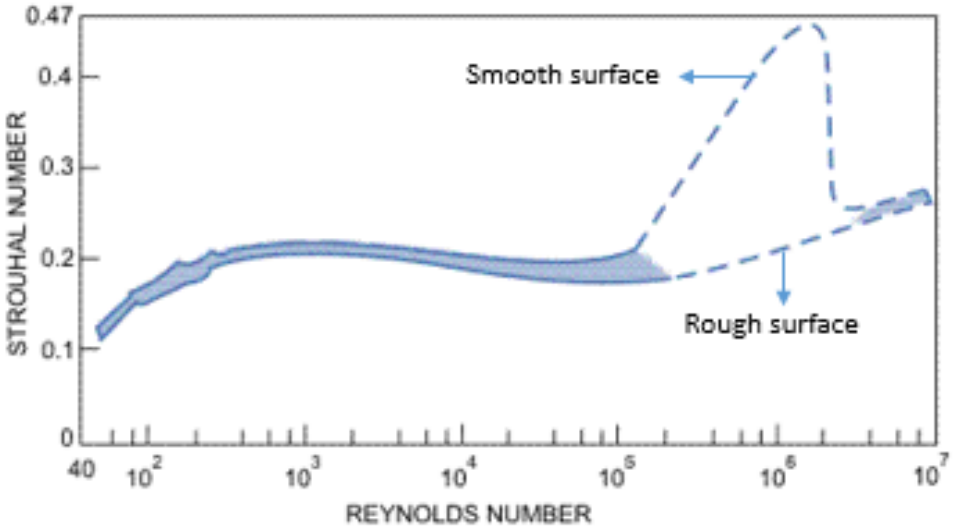


Figure 2.5: Strouhal number – Reynolds number relationship for circular cylinder , originally from Blevins (1990) cited by Sunden (2011)

2.4. Hydrodynamic Forces

Fluid flow over a pipe will create vortex shedding behind the pipe and thus will force the pipe to vibrate in the cross-flow direction and to a certain extent in the in-line direction as well (Koushan, 2009). The vortex shedding appears with a certain frequency, and when this frequency is close to the natural frequency of the pipe it will cause the pipe to vibrate.

The vortex shedding phenomena will result in hydrodynamic forces acting on the pipe. The forces in cross-flow direction consist of added mass and lift force. The lift force appears due to the pressure difference at the top and the bottom of the pipe. On the other hand, the added mass occurs due to an accelerating body that must deflect some volume of surrounding fluid as it moves through it. Conclusively, added mass will not occur if the structure does not move at all. The hydrodynamic force is in-line direction consists of drag force. Drag force occurs due to the pressure difference induced between the upstream and downstream faces of the pipe (Koushan, 2009).

An experiment was performed by Drescher (1956), cited by Sumer and Fredsøe (1997), where the drag and lift forces (F_D and F_L respectively) from the measured pressure distribution is traced (presented in Figure 2.6).

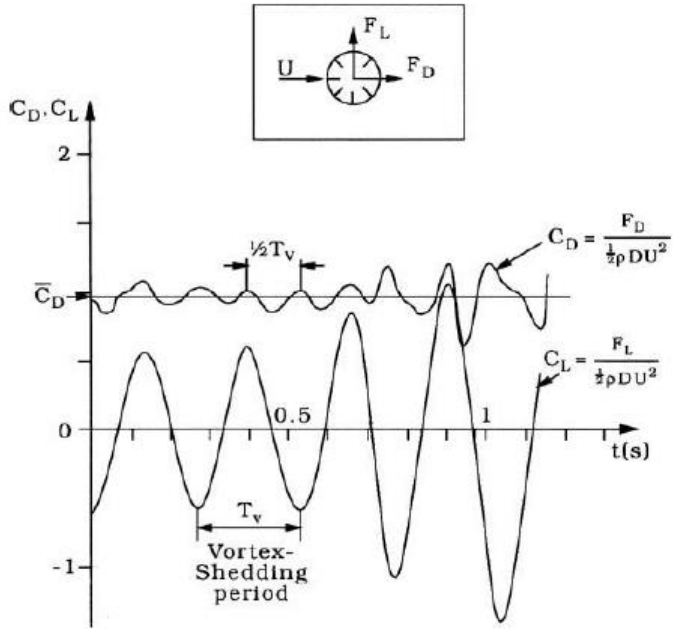


Figure 2.6: Oscillating drag and lift forces time histories (Sumer & Fredsøe, 1997)

According to the time histories in Figure 2.6, Sumer and Fredsøe (1997) stated that:

i. *The force acting on the cylinder in the in-line direction (the drag force) changes periodically in time oscillating around mean drag*
ii. *Even though the incoming flow is completely symmetric with respect to the cylinder axis, there exists a non-zero force component (with zero mean) on the cylinder in the transverse direction (the lift force) and thus this force varies periodically with time as well*

Drag coefficient C_D, mean drag coefficient $\bar{C}_D$, and lift coefficient C_L are the dimensionless parameters for drag and lift forces respectively. It is formulated as

$$C_L = \frac{F_L}{\frac{1}{2} \cdot \rho \cdot L \cdot D \cdot U^2} \tag{5}$$

$$C_D = \frac{F_D}{\frac{1}{2} \cdot \rho \cdot L \cdot D \cdot U^2} \tag{6}$$

$$\bar{C}_D = \frac{\bar{F}_D}{\frac{1}{2} \cdot \rho \cdot L \cdot D \cdot U^2} \quad (7)$$

Where L is the length of the pipe and ρ is the density of the fluid.

2.5. Effect of wall proximity

When a pipe span is formed due to scouring or uneven seabed, there will be a gap between the pipe span and the seabed which is denoted as *e* in Figure 2.7.

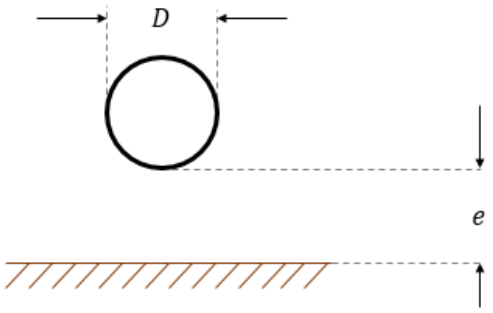


Figure 2.7: The gap between the pipe and seabed

According to Sumer and Fredsøe (1997), when a cylinder is placed near, there will a number of changes occur in the flow around the cylinder. These changes are summarized by Sumer and Fredsøe (1997) as follows:

i. When the gap-ratio (*e/D*) is smaller than 0.3, the vortex shedding will be suppressed.
ii. The stagnation point will move to a lower angular position as seen in Figure 2.8.
iii. The angular position of the separation point (denoted as S in Figure 2.8) will change. The separation point at the free-stream side of the cylinder will move upstream and that at the wall side moves downstream.
iv. The suction is larger on the free-stream side of the cylinder than on the wall-side of the cylinder as seen in Figure 2.9

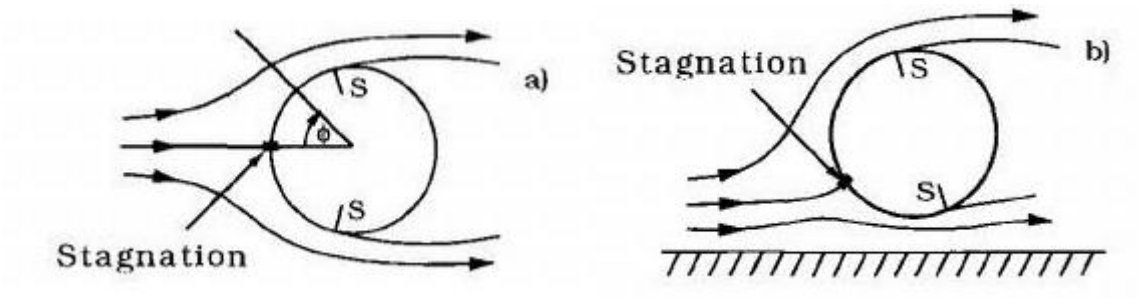


Figure 2.8: Flow around a) free cylinder b) near-wall cylinder (Sumer & Fredsøe, 1997)

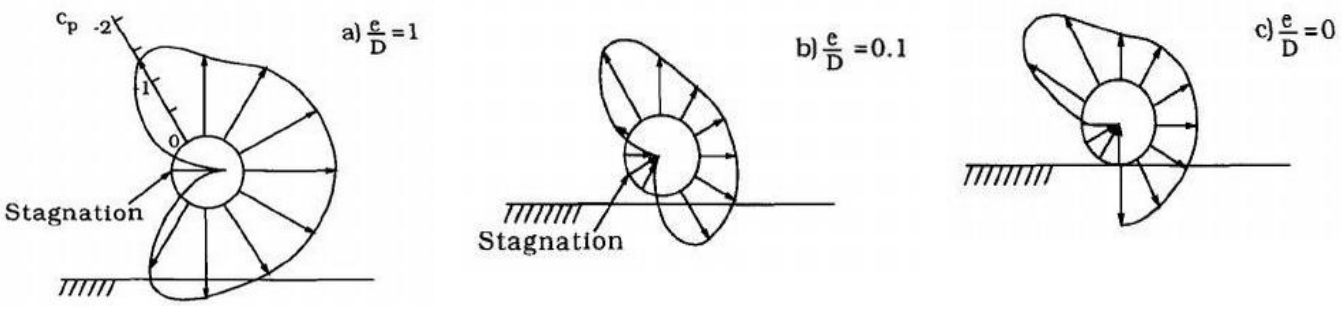


Figure 2.9: Pressure distributions on a cylinder near wall as a function of gap ratio (Sumer & Fredsøe, 1997)

According to Sumer and Fredsøe (1997), the drag coefficient decreases with decreasing gap ratio near wall. This remark is consistent with the pressure distribution that we see in Figure 2.9 where the pressure in horizontal direction is getting smaller as the gap ratio decreases. One characteristic point in the variation of $\bar{C}_D$ is that it increases in a monotonous manner as the gap ratio is increasing to a certain value, and afterwards it remains reasonably constant for further increase in the gap ratio, Sumer and Fredsøe (1997).

The mean flow around the near-wall cylinder is not symmetric, therefore a non-zero mean lift must exist. This remark is contrary to the case of a free cylinder where the mean lift coefficient is always zero. Sumer and Fredsøe (1997) explained that whilst the lift is fairly small for small gap ratios (e/D = 0.2 – 0.3), it increases as the gap ratio is decreased.

2.6. Dynamics of a One Degree of Freedom System

The representation for one degree of freedom system is a spring-mass-damper system (as illustrated in Figure 2.10) where the spring has no damping or mass, the mass has no stiffness or

damping, the damper has no stiffness or mass (Gavin, 2014). Additionally, the mass is allowed to move in one direction only. Based on this statement, thus the vertical vibration of a pipe span in this thesis can be modeled as a one degree of freedom system.

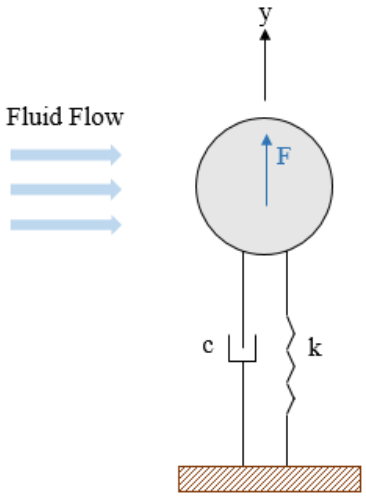


Figure 2.10: Model of a one degree of freedom system in pipe span, adapted from Sumer and Fredsøe (1997)

The dynamic equation of equilibrium or the equation of motion for Figure 2.10 is

$$m\ddot{y}(t)+c\dot{y}(t)+ky=F(t) \tag{8}$$

Where m is the total mass of the pipe, c is the linear-viscous damping, k is the stiffness of the spring, F is the forces acting on the mass points, and y is the vertical displacement of the center of mass of the moving pipe. The dots over y indicates differentiation with respect to time. According to Gavin (2014), the solution to equation (8) is the sum of a homogenous part (free response) and a particular part (forced response).

2.6.1. Free Non-Damped Motions

For free vibrations with no external forces ($F = 0$) and damping ($c = 0$), the equation of motion for this system is as follows

$$m\ddot{y}(t)+ky(t)=0 \tag{9}$$

Un-damped system oscillates freely at its circular natural frequency ω_n, Gavin (2014). In here we introduce the circular Eigen frequency ω_n (note that the unit of ω is radians per second) that is formulated as follows

$$\omega_n = \sqrt{\frac{k}{m}} \tag{10}$$

$$\omega_n \cdot T_n = 2\pi \quad \text{or} \quad T_n = \frac{2\pi}{\omega_n} \tag{11}$$

This leads to the natural frequency f_n as follows

$$f_n = \frac{1}{T_n} = \frac{2\pi}{\omega_n} \tag{12}$$

By substituting equation (10) to equation (9) we obtain

$$\ddot{y}(t) + \omega_n^2 y(t) = 0 \tag{13}$$

Equation (13) has the general solution

$$y(t) = A \cdot \sin(\omega_n t) + B \cdot \cos(\omega_n t) \tag{14}$$

Where A and B are determined by this initial condition: y = 0 for t = 0 → B = 0. And therefore, we get

$$y(t) = A \cdot \sin(\omega_n t) \tag{15}$$

In equation (15), *A* indicates the amplitude of the vibrations

2.6.2. Free Damped Motions

Damping may be introduced into the structure through various mechanisms as described in Figure 2.11. Structural damping is generated due to the friction, impact, and the rubbing movement between the parts of the structure. Material damping is generated due to the internal energy dissipation of materials such as rubber which has a very high material damping. Fluid dynamic damping is generated as the fluid moves relative to the vibrating structure (Sumer & Fredsøe, 1997). In pipeline free span, it is the fluid damping that is dominant.

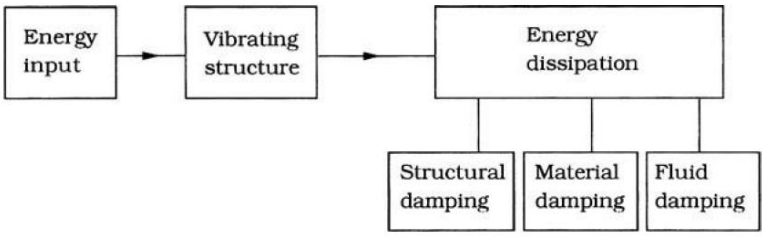


Figure 2.11: Energy input and dissipation on the vibrating structure (Sumer & Fredsøe, 1997)

When the structure is damped, the dynamic response will decay over time. However, Gavin (2014) mentioned that linear viscous damping is complicated to analyze and hence we will limit our attention to linear viscous damping, in which the damping force, $f_{damping}$, is proportional to the velocity.

$$f_{damping} = c\dot{y} \tag{16}$$

Therefore, by assuming that the system is damped and not subjected to any external force (F = 0), the equations of motion of this system is

$$m\ddot{y}(t) + c\dot{y}(t) + ky(t) = 0 \tag{17}$$

Assuming the solutions is in the form of

$$y(t) = C \cdot e^{st} \tag{18}$$

According to Gudmestad (2014), we will obtain the "characteristic equation" from equation (17) for the exponents as following

$$s^2 + \left(\frac{c}{2m}\right)s + {\omega_n}^2 = 0 \tag{19}$$

Equation (19) has the solutions

$$s_{1,2} = \left(\frac{c}{2m}\right) \pm \sqrt{\left(\frac{c}{2m}\right)^2 - {\omega_n}^2} \tag{20}$$

In order to make the solution of equation (19) represents an oscillating motion, the expression under the square root sign must be negative (Gudmestad, 2014). Therefore, it gives

$$\frac{c}{2m} < \omega_n \quad \text{or} \quad c < 2 \cdot m \cdot \omega_n \tag{21}$$

If $c = 2 \cdot m \cdot \omega_n$, Gudmestad (2014) said that the damping is a "critical damping" and hence the term "relative damping", λ, is introduced

$$\lambda = \frac{c}{2m\omega_n} \tag{22}$$

The term λ represents the relation between the actual damping and the critical damping. Normally, damping is expressed by the relative damping (Gudmestad, 2014). Depending on the value of the damping, there are three different cases as follows:

i. Under critical damping: $\lambda < 1$

When the relative damping is positive and less than 1, the system will oscillate freely from initial displacement y_0 and velocity (Gavin, 2014). This kind of case is the normal case for structures (Gudmestad, 2014). Introducing this damping gives the solution

$$y(t) = e^{-\lambda\omega_n t}\left(A\cdot\sin(\omega_d t) + B\cdot\cos(\omega_d t)\right) \tag{23}$$

A and *B* are constants, determined by the boundary condition at t = 0. The damped frequency of oscillations, ω_d, is given by

$$\omega_d = \omega_n\sqrt{1-\lambda^2} \tag{24}$$

The relative damping determines how fast the oscillations are damped. The solution for underdamped system represents exponentially decreasing oscillations, as seen in Figure 2.12. The damping ratio for an underdamped system in the vertical displacement time histories (Figure 2.12) is the natural log of the amplitudes of any two successive peaks (Gudmestad, 2014).

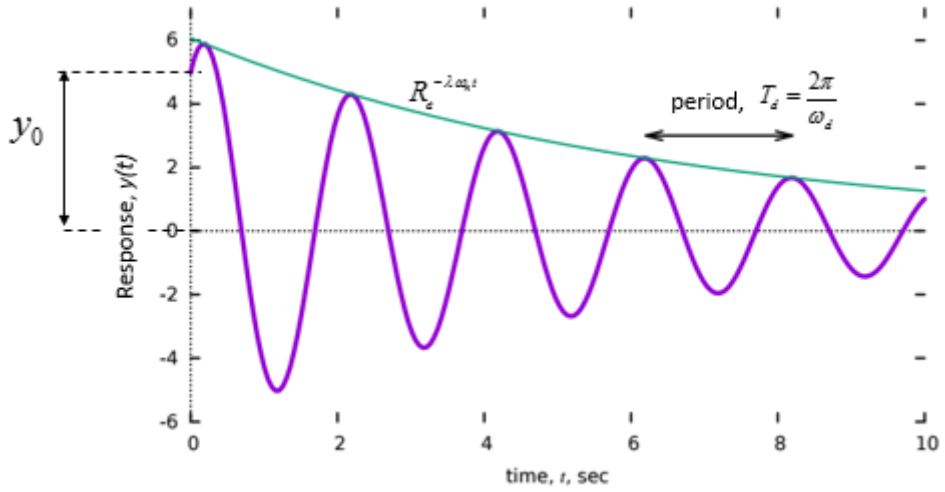


Figure 2.12: Under damped system oscillations, adapted from Gavin (2014) and Gudmestad (2014)

ii. Critical damping: $\lambda = 1$

Critical damping means that $c = 2\cdot m\cdot\omega_n$ and hence the two roots of equation (20) coincide as follows

$$s_1 = s_2 = -\left(\frac{c}{2m}\right) = -\omega_0 \tag{25}$$

The solution for this system's motion equation becomes

$$y(t) = (A + Bt)e^{-\omega_n t} \tag{26}$$

Similarly, *A* and *B* are constants that are determined by the boundary condition at t = 0. Critically-damped system has no oscillation; rather it has a decaying motion as shown in Figure 2.13, Gudmestad (2014).

i. Over critical damping: $\lambda > 1$

If the damping is greater than the critical damping, it results in the roots of the equation (20) to be distinct and real. When a system is over-damped, it will not oscillate freely (Gavin, 2014). The solution for this system's motion equation becomes

$$y(t) = e^{-\lambda\omega_n t}\left(Ae^{-\omega_d t} + Be^{-\omega_d t}\right) \tag{27}$$

Similarly, *A* and *B* are constants that are determined by the boundary condition at t = 0. For over-damped systems, we will obtain an exponentially decreasing motion as shown in Figure 2.13.

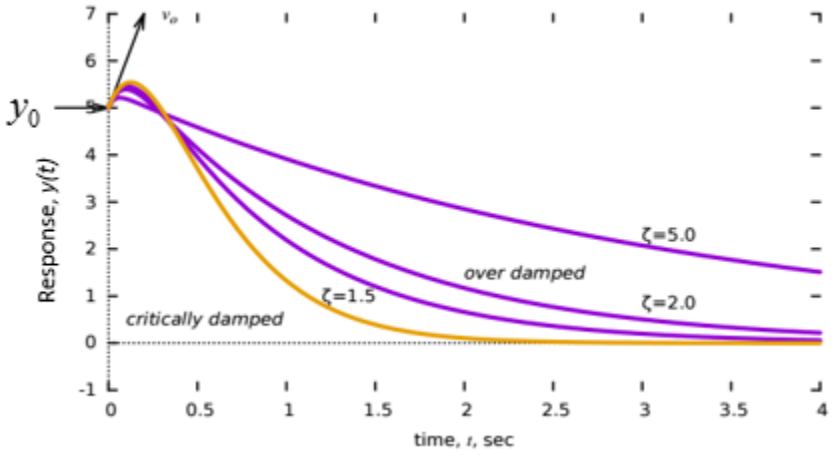


Figure 2.13: Free response of critically-damped (orange) and over-damped (purple) oscillators to an initial displacement, adapted from Gavin (2014)

2.6.3. Forced Oscillations with Viscous Damping

If a one degree of freedom system is subjected to a load, then the solution for the oscillations is given by the sum of a homogenous and a particular solution.

$$y = y_h + y_p \tag{28}$$

The homogeneous solution y_h is given by equation (23). The solution may have large numerical values for small *t*, but this part of the total solution is dampened out with time (Gudmestad, 2014). On the other hand, the particular solution, y_p, will last as long as there is an external load.

In this thesis, the external load is represented by the harmonic load caused by the vortex shedding phenomena. Additionally, we should note that other periodic load such as wave load is also relevant for this case. All periodic motion can be given as a sum of harmonic functions through Fourier expansions (Gudmestad, 2014). The loading term is in the form of

$$F(t) = F_0 \cdot \sin(\omega t) \tag{29}$$

If the harmonic load is persistent, then after several cycles the system will respond only at the frequency of the external force, ω (Gavin, 2014). The particular solution then is in the form of as follows

$$y_p(t) = y_0 \sin(\omega t - \theta) \tag{30}$$

The initial amplitude y_0 is given by

$$y_0 = \frac{F_0}{m\omega_n^2} DAF = \frac{F_0}{k} DAF \tag{31}$$

Where the dynamic amplification factor, *DAF*, is given by

$$DAF = \left(\left(1-\beta^2\right)^2 + \left(2\lambda\beta\right)^2 \right)^{-\frac{1}{2}} \tag{32}$$

And the relative frequency relation, β, which represents the relation between the frequency ω of the loading and the angular Eigen frequency, ω_n, is given by

$$\beta = \frac{\omega}{\omega_n} \tag{33}$$

The dynamic amplification factor, *DAF*, states how much the dynamic response is, as compared to the static response causes by the static loading, F_0 (Gudmestad, 2014). The phase angle, θ, is the angle between the loading *F(t)* and the response *y(t)*.

$$\theta = arctg\left(\frac{2\lambda\beta}{1-\beta^2} \right) \tag{34}$$

The dynamic amplification factor, *DAF*, and the phase angle, θ, are shown in Figure 2.14 and Figure 2.15.

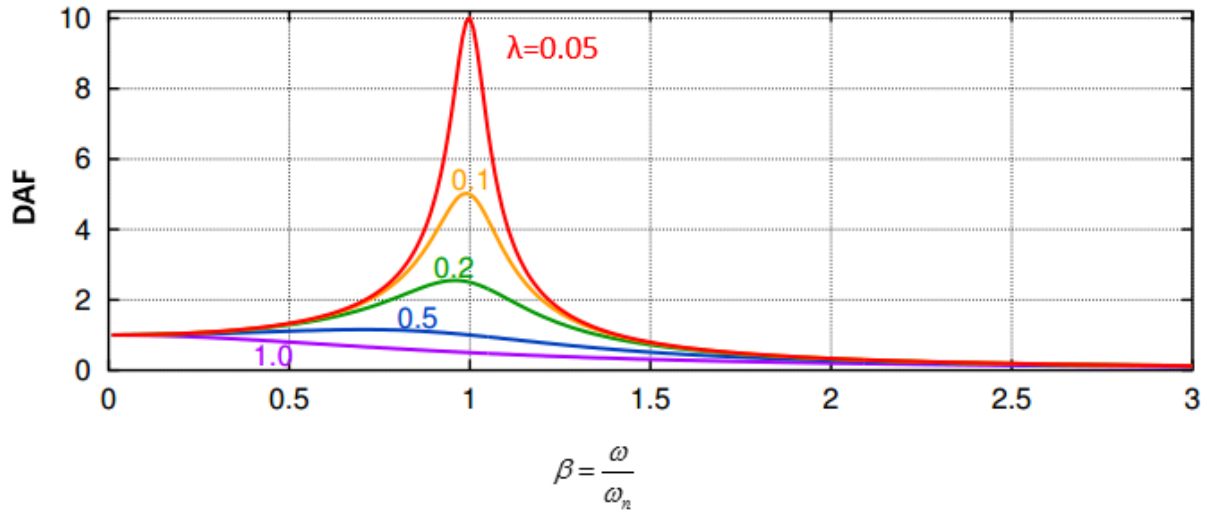


Figure 2.14: Dynamic amplification factor as a function of relative frequency, adapted from Gavin (2014) and Gudmestad (2014)

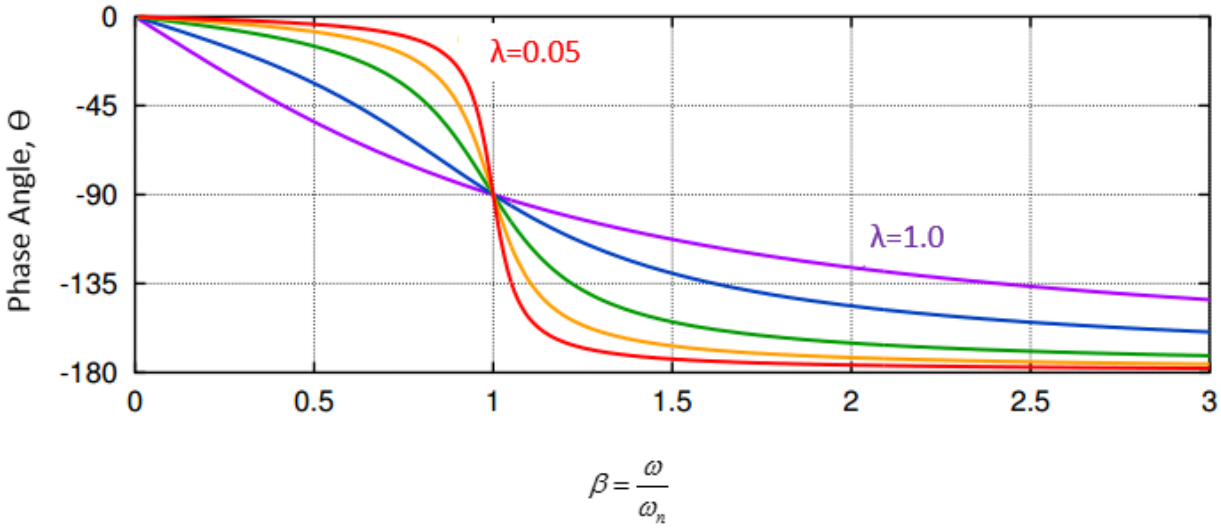


Figure 2.15: The phase angle as a function the relative frequency, adapted from Gavin (2014) and Gudmestad (2014)

Figure 2.15 shows that the dynamics can be divided into three different cases. As quoted from Gudmestad (2014), those cases are as follows:

- *β << 1: The motion or dynamics are controlled by the stiffness of the system and is in phase with the loading (for very stiff structures). We have little or no dynamic effects; static analysis will suffice.*

- *$\beta \approx 1$: The motion is controlled by the damping in the system, and is 90 degrees out of phase with the loading (resonant motion). We have large dynamic effects, determined by the damping; we shall try to avoid this situation.*
- *$\beta >> 1$: The motion is controlled by the mass and is in phase to the loading (180 degrees), that is, mass controlled forced vibration. The mass or inertia force acts against the loading and reduces the response.*

The homogenous part of the total solution of equation (28) will subside gradually and then the total solution will asymptotically approach the particular solution in equation (30), as illustrated in Figure 2.16.

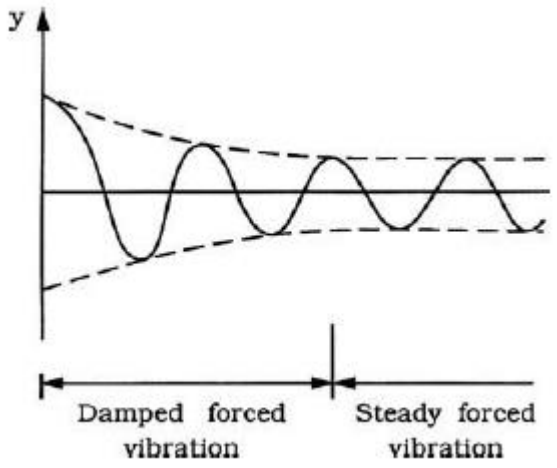


Figure 2.16: Forced vibrations with viscous damping (Sumer & Fredsøe, 1997)

2.7. Reduced velocity

The distance which the undisturbed flow is traveling during one cycle, *U/f*, defines the path length per cycle for steady vibrations (Koushan, 2009). The reduced velocity, V_R, indicates the ratio of the path length per cycle to the body width (which in this case is represented by the cylinder's diameter, *D*). According to Karunakaran (2014), the reduced velocity is formulated as follows

$$V_R = \frac{U}{f_0 \cdot D} \tag{35}$$

In the formula, f_0 is the Eigen frequency of the pipe in still water (free oscillation test, will be explained more in section 2.8), *U* is the flow velocity, and *D* is the diameter of the pipe.

An experiment was done by Feng (1968) where he mounted a D-section cylinder on vertical springs so that the system only has one degree of freedom and then exposed it to an increased air flow with small increments. Parameters such as vortex shedding frequency f_v, the vibration/oscillation frequency f_{osc}, the amplitude of the oscillation A, and also the phase angle θ were measured in his experiment. The results of his experiments are presented in Figure 2.17.

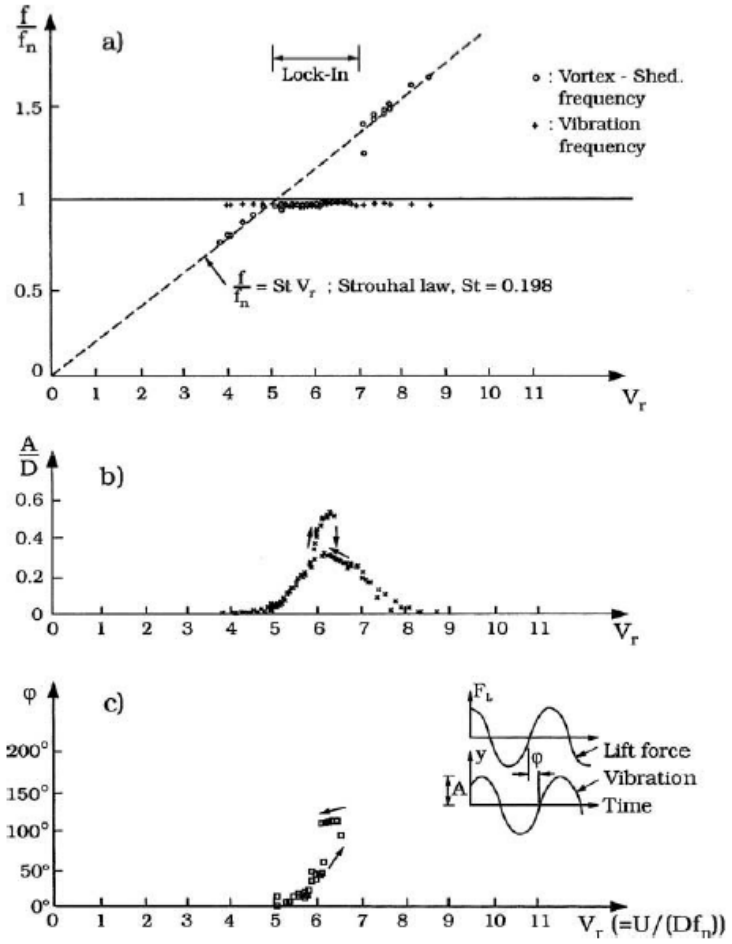


Figure 2.17: Feng's experiment responses (Feng, 1968)

According to Karunakaran (2014), the cross-flow oscillations begin when the reduced velocity V_R is about 3, it reaches peak when V_R is about 5, and continue to higher V_R. Afterwards, in the range of $5 < V_R < 7$, the vortex shedding frequency and Eigen frequency lock on to each other; this is known as the "lock-in" phenomenon. At this range, the vortex shedding frequency and the oscillation frequency collapse into the natural frequency of the system. Conclusively, lock-in is also a response that indicates resonance.

We shall note that in lock-in phenomena the vortex shedding frequency and Eigen frequency lock-on to each other, but Eigen frequency will be influenced from the change of added mass and the oscillation itself will influence the vortex shedding frequency (Karunakaran, 2014). Additionally, the in-line oscillations begin when the reduced velocity is about 1.5, but are often small enough to not be concerned.

2.8. Added Mass from Free Oscillation

When a free oscillation is surrounded by fluid, the hydrodynamic force will be in phase with the acceleration and contribute to added mass. In the free oscillation test, added mass can be found from the measured force. The mass that is used to calculate angular Eigen frequency, ω_n, shall also include the added mass into its calculation. Therefore, equation (10) can also be re-written as follows

$$\omega_n = \sqrt{\frac{k}{m}} = \sqrt{\frac{k}{m_{dry} + m_a}} \tag{36}$$

Where the m_{dry} is the weight of the pipe in the air and m_a is the added mass.

$$m_a = \frac{\pi D^4}{4} \cdot \rho \cdot C_a \cdot L \tag{37}$$

An analytical added mass has been already defined by Det Norske Veritas (2011) in DNV-RP-H103. According to Det Norske Veritas (2011), the analytical added mass per unit length, A_{ij}, for 2-dimensional body, i.e. long cylinder in an infinite fluid (far from boundaries), is formulated as follows

$$A_{ij} = \rho \cdot C_A \cdot A_R \tag{38}$$

Where C_A is the analytical added mass coefficient, ρ is the density of the fluid, and A_R is the reference area. The value of C_A for circular section through body is defined as 1 (Det Norske Veritas, 2011). The reference area for circular section is formulated as follows

$$A_R = \pi \cdot \left(\frac{D}{2}\right)^2 \tag{39}$$

CHAPTER 3 - Computational Fluid Dynamics

Computational Fluid Dynamics (CFD) is a very powerful analytical technique that involves fluid flow, heat transfer and associated phenomena such as chemical reactions by means of computer-based simulation (Versteeg & Malalasekera, 2007). CFD is developed to provide a capability that is comparable with other computer-aided engineering tools such as stress analysis codes. CFD codes can produce extremely large amounts of results with virtually no added expense. It is indeed very cheap compared with performing parametric studies.

3.1. CFD Methodology

All commercial CFD packages include sophisticated user interfaces to input problem parameters and to examine results which result in an easy access to their solving problems (Versteeg & Malalasekera, 2007). Therefore, all CFD codes consist of three main elements as described in Figure 3.1

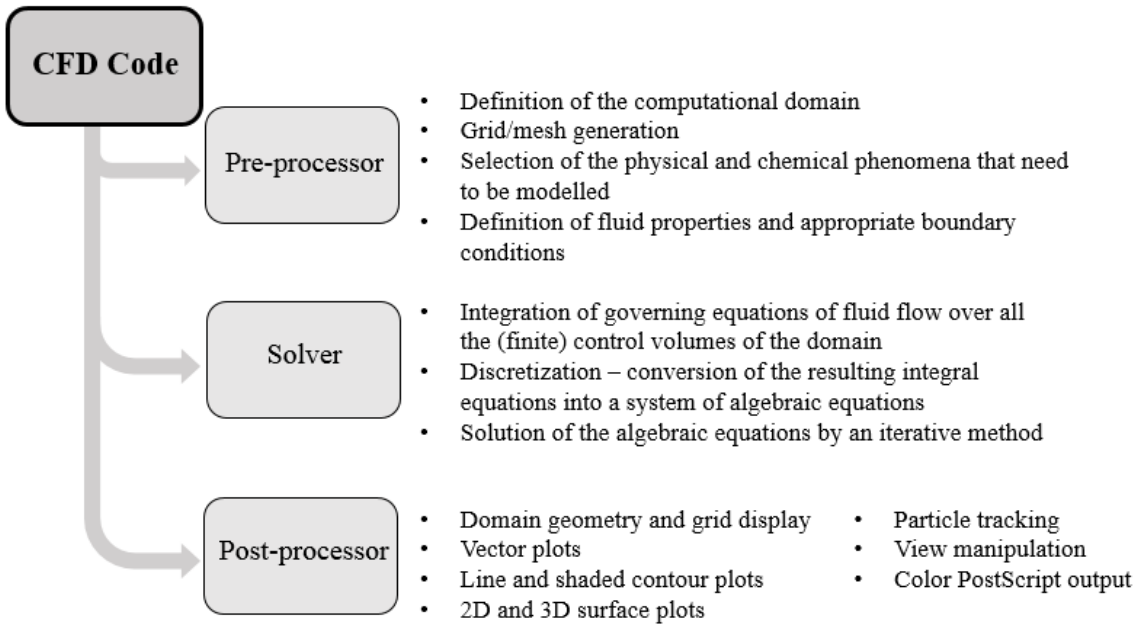


Figure 3.1: Summary of elements in CFD codes, adapted from Versteeg and Malalasekera (2007)

i. Pre-processor

At this stage, the geometry of the region of interest is generated as the computational domain and then it will be divided into a number of smaller, non-overlapping sub-domain: a grid (or mesh) of discrete cells (Versteeg & Malalasekera, 2007). The accuracy of a CFD solution is governed by the number of cells in the grid. Therefore, over 50% of the time spent on a CFD simulation is usually devoted to the definition of the domain geometry and grid generation. The fluid properties and boundary conditions are also defined at this stage.

ii. Solver

There are three distinctive numerical solution techniques in CFD, which are as follows:

- The finite difference method (FDM): one of the easiest methods to use, especially for simple geometries (Asyikin, 2012). Nowadays, there exists a higher-order FDM (Moin, 1991) that is widely used for problems with complex geometries
- The finite element method (FEM): commonly used for structural analysis, but also applicable for fluids. In FEM, the domain will be divided into unstructured discrete volumes (finite elements).
- The finite volume method (FVM): the most common approach that is used in CFD codes. This method accommodates any type of grid and hence it is also suitable for complex geometries

Now we will consider only the FVM as it is the finite difference formulation that is used in the most well-established CFD codes. The finite volume method divides the solution domain into a finite number of contiguous control volume (CV) and the conservation equations are applied to each control volume (Asyikin, 2012). This is the aspect that distinguishes the FVM from all other CFD numerical techniques. According to Versteeg and Malalasekera (2007), the conservation of a general variable ϕ within a finite control volume can be expressed as

Rate of change of ϕ in the control volume with respect to time	=	*Net rate of increase of ϕ due to convection into the control volume*	+	*Net rate of increase of ϕ due to diffusion into the control volume*	+	*Net rate of creation of ϕ inside the control volume*

iii. Post Processor

Massive development works have recently taken place in the post-processing nowadays. The leading CFD packages are equipped with versatile data visualization tools due to the increasing popularity of engineering workstations, which many of it has outstanding graphics capabilities (Versteeg & Malalasekera, 2007).

3.2. Governing Equations

The CFD problems are governed by a set of mathematical equations and solved numerically. The mathematical equations are based on the conservation laws physic which are the governing equations for fluid flow. For fluid flow problem, we need 4 governing equations to solve the velocities (for both 3 directions) and the pressure. The governing equations would be the mass balance and 3 momentum equations.

3.2.1. Conservation of Mass

The mass conservation theory states that the mass will remain constant over time in a closed system. According to Versteeg and Malalasekera (2007), the first step in the derivation of the mass conservation is to write down a mass balance for the fluid element:

Rate of increase of mass in fluid element	=	Net rate of flow of mass into fluid element

The mass conservation equation, which is also called as the mass balance equation, is written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \cdot \vec{u} \right) = 0 \tag{40}$$

The mass balance equation applies for unsteady, three-dimensional mass conservation or continuity at a point in a compressible fluid. If we assume the flow is in steady state, which means

$$\frac{\partial}{\partial t} = 0 \tag{41}$$

And the fluid is incompressible, where the density ρ is constant, the mass balance becomes

$$\frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} + \frac{\partial (\rho w)}{\partial z} = 0 \tag{42}$$

Or in vector notation is written as

$$\nabla \cdot \vec{u} = 0 \tag{43}$$

3.2.2. Conservation of Momentum

The conservation of momentum is originally expressed in Newton's 2nd law. According to Versteeg and Malalasekera (2007), the rate of change of momentum of a fluid particle equals the sum of the forces on the particle:

$$\begin{matrix}\text{Rate of increase of} \\ \text{momentum of fluid particle}\end{matrix} = \begin{matrix}\text{Sum of forces on} \\ \text{fluid particle}\end{matrix}$$

The momentum conservation equation in x-, y-, and z-direction are written as follows

x-direction: $$\rho\frac{Du}{Dt} = \frac{\partial\left(-p+\tau_{xx}\right)}{\partial x} + \frac{\partial\tau_{yx}}{\partial y} + \frac{\partial\tau_{zx}}{\partial z} + S_{Mx} \quad (44)$$

y-direction: $$\rho\frac{Dv}{Dt} = \frac{\partial\tau_{xy}}{\partial x} + \frac{\partial\left(-p+\tau_{yy}\right)}{\partial y} + \frac{\partial\tau_{zy}}{\partial z} + S_{My} \quad (45)$$

The viscous stresses are denoted by τ where the suffix i and j in τ_{ij} indicate that stress component acts in the *j*-direction on a surface normal to *i*-direction. The source S_M is defined as contribution to the body forces in the total force per unit volume on the fluid.

3.2.3. Navier-Stokes Equations for a Newtonian Fluid

The Navier-Stokes equation is used as the governing equation for an incompressible and laminar flow of a Newtonian fluid. In a Newtonian fluid, the viscous stresses are proportional to the rates of deformation (Versteeg & Malalasekera, 2007). According to Hjertager (2015), the Newton's law of viscosity is as follows

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\cdot\frac{\partial u_i}{\partial x_i}\cdot\delta_{ij} \quad (46)$$

Where $\mathcal{S}$ denotes the rate of linear deformation of a fluid element. The suffix system of δ is identical with τ. If we insert the Newton's law of viscosity into the momentum of balance, we will get the Navier-Stokes equations as follows:

x-direction:

$$\rho\frac{Du}{Dt} = -\frac{\partial p}{\partial x} + div\left(\mu\cdot\nabla u\right) + S_{Mx} \quad (47)$$

y-direction:

$$\rho\frac{Dv}{Dt} = -\frac{\partial p}{\partial y} + div\left(\mu\cdot\nabla v\right) + S_{My} \quad (48)$$

3.3. General Transport Equation

It can be seen from section 3.2 that there are some similarities between the governing equations. According to Versteeg and Malalasekera (2007), if we introduce a general variable ϕ, the conservative form of all fluid flow equations, including equations for scalar quantities such as temperature, can be usefully written as transport equation for property ϕ

$$\frac{\partial(\rho\phi)}{\partial t} + div(\rho\vec{u}\phi) = div(\Gamma \cdot grad\phi) + S_\phi \tag{49}$$

In the formula, Γ is the diffusion coefficient and S_ϕ is the source inside the fluid element. Additionally, we shall note that there is no transport equation for pressure. The terms in the left hand side of the transport equation are the rate of change term (transient term) and the convective term. The terms in the right hand side of the transport equation are the diffusive term and the source term. In steady state problems, the rate of change term is equal to zero, and thus the transport equation becomes

$$div(\rho\vec{u}\phi) = div(\Gamma \cdot grad\phi) + S_\phi \tag{50}$$

Transport equation is used as the starting point for computational procedures in the finite volume method. Therefore, the transport equation is to be integrated over a three-dimensional control volume as it is the key step of the finite volume method.

3.4. Solution Algorithms for Pressure-Velocity Coupling in Steady Flows

It was stated by Versteeg and Malalasekera (2007) that the transport equations for each velocity component – momentum equations – can be derived from the general transport equation by replacing ϕ by *u*, *v*, and *w* respectively. The 3 momentum equations and continuity equation are coupled because every velocity component appears in each momentum equation and in the continuity equation. The most concerned issue that needs to be resolved is the role played by the pressure. Pressure appears in both momentum equations, but there is no transport or other equation for the pressure.

If the flow is incompressible (constant density), there is no equation linked to the pressure. In this case, coupling between the pressure and velocity will introduce a constraint in the solution of the flow field. The pressure-velocity coupling equation can be solved by introducing a particular

numerical procedures called the solution algorithms. There are many algorithms that have been developed until recently, such as SIMPLE (steady-state), SIMPLEC, SIIMPLER, and PISO (transient problem).

3.5. Turbulence Modelling

Turbulent flow has irregular behaviors of a flow where various quantities such as velocity and pressure show random variation in time and space coordinates. A typical point velocity measurement of turbulent flow exhibits the form shown in Figure 3.2.

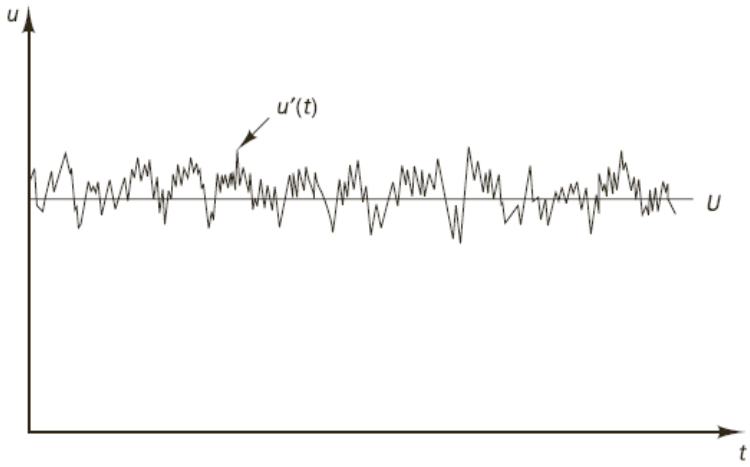


Figure 3.2: Point measurement velocity in turbulent flow (Versteeg & Malalasekera, 2007)

The velocity in the turbulent flow will be decomposed into a steady mean value $\bar{U}$ with a fluctuating component *u'(t)* superimposed on it; this is called as the Reynolds decomposition (Versteeg & Malalasekera, 2007).

$$u(t) = \bar{U} + u'(t) \tag{51}$$

The same principle applies for pressure and others

$$\varphi(t) = \bar{\varphi} + \varphi'(t) \tag{52}$$

Where φ denotes the variable quantity that varies in random behavior in turbulent flow. A turbulence model is a computational procedure close to system of mean flow equations. It allows the calculation of the mean flow without first calculating the full time-dependent flow field. In particular, we need expressions for the Reynold stresses (Bakker, 2006b).

There are 3 numerical method approaches for turbulence models which are direct numerical simulation (DNS), large eddy simulation (LES), and Reynolds-averaged Navier Stokes (RANS). In short, the difference between RANS, DNS, and LES is described in the Figure 3.3

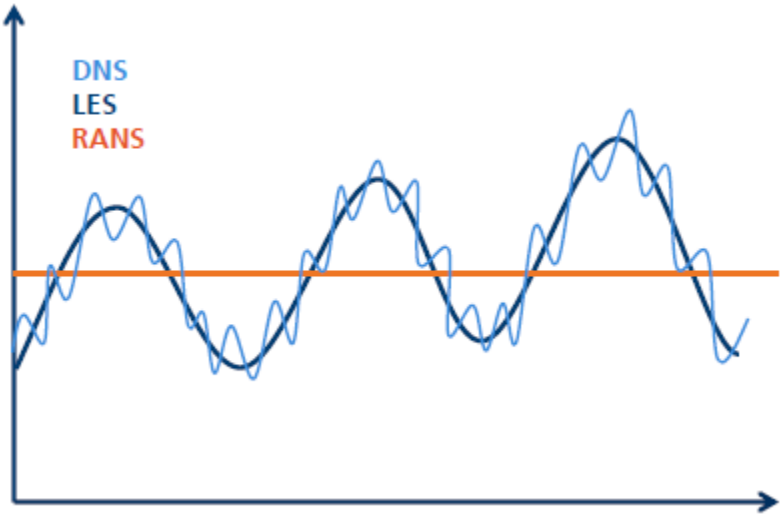


Figure 3.3: Comparison of DNS, LES, and RANS model (Giljarhus, 2015)

3.5.1. Reynolds-averaged Navier Stokes Turbulence model

Reynolds-averaged Navier Stokes (RANS) equations are the most classical approach to turbulence modeling. According Versteeg and Malalasekera (2007), RANS equation is focused on the mean flow and the effects of turbulence on mean flow properties. The Navier-Stokes are time averaged before the numerical methods are applied. Hence extra terms (transport equations) will appear in the time-averaged (or known as Reynolds-averaged) flow equations because of the interactions between various turbulent fluctuations. These extra terms will be solved using classical turbulence models that are listed in Table 3.1.

Table 3.1: RANS turbulence models, Versteeg and Malalasekera (2007)

No. of extra transport equations	Name
0	Mixing length model
1	Spalart-Allmaras model
2	$k-\varepsilon$ model, $k-\omega$ model, Algebraic stress model
4	Reynolds stress model

According to Symscape (2009), the advantage of using the RANS equations for steady fluid low simulation is that the mean flow velocity is calculated as a direct result without the need to average the instantaneous velocity over a series of time steps.

3.5.1.1. The $k-\varepsilon$ model

K-epsilon turbulence model is a sophisticated and general turbulence model that is used in Computational Fluid Dynamics (CFD) that allows for the effect of transport of turbulence properties by convection and diffusion and for production and destruction of turbulence. It is a two equation model that gives description of turbulence by means of two transport equations (PDEs), one for the turbulent kinetic energy, k, and the other one for the rate of dissipation of turbulent kinetic energy, ε. According to Versteeg and Malalasekera (2007), the principal assumption of this turbulence model is that the turbulent viscosity is isotropic, which means that the ratio between Reynolds stress and mean rate of deformation is the same in all directions. However, this assumption fails in many complex flows where it leads to inaccurate predictions.

According to Versteeg and Malalasekera (2007), the summary of the performance assessment for the standard k-ε is listed as follows

- Advantages
    - The simplest turbulence model since on initial and/or boundary conditions need to be specified
    - Gives excellent performance in many industrially relevant flows
    - Well established
    - The most widely validated turbulence model
- Disadvantages
    - It could be costly compared with the other turbulence model such as mixing length model
    - Gives poor performance in a variety of important cases such as:
        - Unconfined flows
        - Flows with large extra strains (e.g. swirling flows, curved boundary layers)
        - Rotating flows
        - Flows driven by anisotropy of normal Reynolds stresses (e.g. fully developed flows in non-circular ducts)

3.5.1.2. The SST $k-\omega$ model

According to Versteeg and Malalasekera (2007), Menter (1992) stated that the results of the k-epsilon model are greatly less sensitive to the assumed values in the free stream. However, its near-wall performance is deemed as unsatisfactory for boundary layers with adverse pressure gradients. Therefore, Versteeg and Malalasekera (2007) mentioned that this condition led to a suggestion to use a hybrid model that enables:

(i) An automatic transformation of the k-epsilon model into k-omega model in the near-wall region

(ii) An automatic transformation to the standard k-epsilon model in the fully turbulent region

In short, k-omega SST model combines the better of two worlds and this is why SST model has become popular. The transport equation for *k* for turbulent flows at high Reynolds is as follows

$$\underbrace{\frac{\partial(\rho k)}{\partial t}}_{\text{(I)}}+\underbrace{div\left(\rho k\vec{U}\right)}_{\text{(II)}}=\underbrace{div\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)grad\left(k\right)\right]}_{\text{(III)}}+\underbrace{P_k}_{\text{(IV)}}-\underbrace{\beta^*\rho k\omega}_{\text{(V)}} \tag{53}$$

where

$$P_k=\left(2\mu_t S_{ij}\cdot S_{ij}-\frac{2}{3}\rho k\frac{\partial U_i}{\partial x_j}\delta_{ij}\right) \tag{54}$$

In k-omega SST model, the ε-equation is transformed into an ω-equation by substituting this following formula

$$\varepsilon=k\cdot\omega \tag{55}$$

According to Versteeg and Malalasekera (2007), yhe transformation thus yields the transport equation for ω as follows

$$\underbrace{\frac{\partial(\rho\omega)}{\partial t}}_{\text{(I)}}+\underbrace{div\left(\rho\omega\vec{U}\right)}_{\text{(II)}}=\underbrace{div\left[\left(\mu+\frac{\mu_t}{\sigma_{\omega,1}}\right)grad\left(\omega\right)\right]}_{\text{(III)}}+\underbrace{\gamma_2\left(2\rho S_{ij}\cdot S_{ij}-\frac{2}{3}\rho\omega\frac{\partial U_i}{\partial x_j}\delta_{ij}\right)}_{\text{(IV)}}-\underbrace{\beta_2\rho\omega^2}_{\text{(V)}}+\underbrace{2\frac{\rho}{\sigma_{\omega,2}\omega}\frac{\partial k}{\partial x_k}\frac{\partial\omega}{\partial x_k}}_{\text{(VI)}} \tag{56}$$

In words, according to Versteeg and Malalasekera (2007), the transport equations are expressed as

Rate of change of k or ω	+	*Transport of k or ω by convection*	=	*Transport of k or ω by turbulent diffusion*	+	*Rate of production of k or ω*	-	*Rate of dissipation of k or ω*	+	*Cross diffusion term which arises during the* $\varepsilon = k \cdot \omega$ *transformation*
(I)		(II)		(III)		(IV)		(V)		(VI)

3.5.2. Large Eddy Simulation

The operation concept of LES is low-pass filtering. It means that the small scales of the transport equation solution are taken out by applying the low-pass filtering. LES reduces the computational cost of the simulation because only large eddies, which contain most of the energy, are resolved. LES enables new areas of application such as vibration and aero-acoustics where the velocity fluctuations play an important role (Giljarhus, 2015).

3.5.3. Direct Numerical Simulation

DNS is a method to predict the turbulent flow where the Navier-Stokes equations are numerically solved without averaging. In short, all of the turbulent motions in the flow are solved. Consequently, DNS requires a very high computational cos, even if it is used to perform simulation at low Reynolds number. DNS is a useful tool in fundamental research in turbulence, but it is rarely used due to the limitation of the processing speed and the memory of the computer.

3.6. Introduction to OpenFOAM

OpenFOAM (Open Field Operation and Manipulation) is an open source, Linux based, free CFD software. OpenFOAM has collections of solvers that can be developed and applied to various complex fluid flow problems such as chemical reactions, turbulence and heat transfer, etc. The capabilities of OpenFOAM can be compared to almost all commercial applications used to resolve CFD problems in general. A plugin in OpenFOAM called paraFOAM can visualize the data and the results of the mesh in `paraView`.

OpenFOAM provides a lot of solvers, utilities, and libraries that can be configured and developed just like the other CFD software does. Every case file in OpenFOAM consists of 3 folders as shown in Figure 3.4.

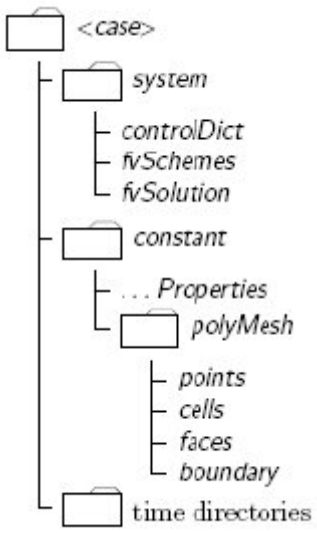


Figure 3.4: OpenFOAM file structures (Greenshields, 2015)

The `system` directory has at least 3 files inside it; `fvSolution` that determines which solver that will be used, `fvSchemes` to configure the discretization scheme, and `controlDict` to control the simulation time and information storage.

Inside the `constant` directory, there resides a subdirectory `polyMesh` that consists of `blockMeshDict`, `faces`, `owner`, `neighbor`, and `points` which are the property of preprocessing result. There are also another files inside the `constant` directory which will be needed to define the simulation properties such as turbulence (e.g. `turbulenceProperties`) and transport models (`transportProperties`).

Time directories include the data for every time interval that has been set up in `controlDict`. Before proceeding with the simulation, an initial time directory named as `0` must be made. This directory shall include the properties of initial or boundary condition.

3.6.1. Mesh generation

In the OpenFOAM, there are many ways available to generate a mesh. One of them is by using a dictionary file named `blockMeshDict` located in the `constant/polyMesh` directory. After the `blockMeshDict` is set up, the command `blockMesh` in the terminal will read the dictionary, generate the mesh, and write out the mesh data to points and faces, cells and boundary files, in the

same directory (CFD Direct, 2015). According to CFD Direct (2015), the concept of `blockMesh` is to decipher the domain geometry into a set of 1 or more three dimensional, hexahedral blocks. The edges of the blocks can be configured as straight lines, arcs, or splines. Each block of the domain geometry consists of 8 vertices, one at each corner of a hexahedron. The vertices will be written as a list so that later on it can be assessed using its label by the OpenFOAM.

3.6.2. Assessment of Mesh Quality

Mesh generation is the most important and time consuming stage in the CFD simulation. The reason of its importance is it has significant impact on the rate of convergence, solution accuracy, and the time required by CPU to do the simulation (Bakker, 2006a). According to B. Hjertager (2009) and Bakker (2006a), a mesh structure (or domain, interchangeably) consists of points (vertices), faces, cells, edges, and patches. An illustration of mesh structures in CFD is presented in Figure 3.5.

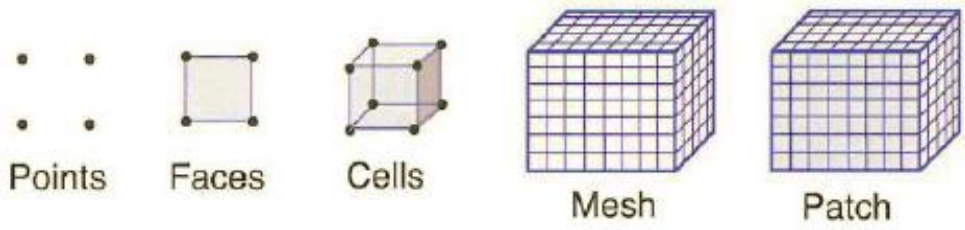


Figure 3.5: Illustration of mesh structures, B. Hjertager (2009)

There are many different cell/element and grid types available in the CFD. The selection depends on the problem and the solver capabilities. According to Bakker (2006a), the cell/element types are divided into 2 major classes; 2-dimensional and 3-dimensional. Each class has its own cell/element types that are suitable for the simulation (refer to Figure 3.6).

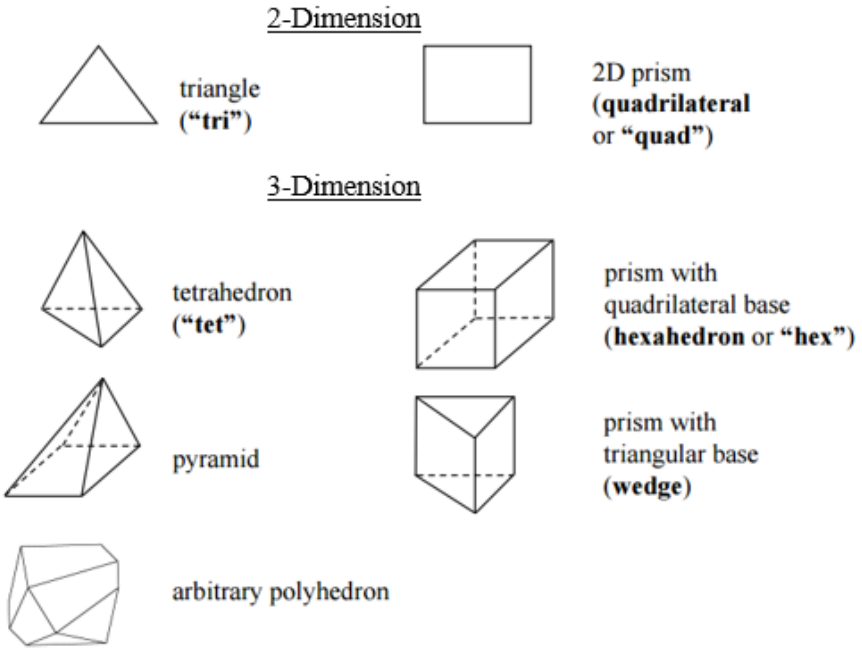


Figure 3.6: Examples of cell types (Bakker, 2006a)

In grid/mesh generation, we should always strive for a high quality mesh since a poor quality mesh will cause an inaccurate solution and convergence. According to Bakker (2006a), the quality of the mesh is assessed by observing the parameters as follows:

1. Skewness

   Skewness determines how ideal a face or a cell is. Typically, highly skewed faces and cells are not acceptable because initially the equations are being solved by assuming that the cells are relatively equilateral/equiangular as illustrated in Figure 3.7 (Asyikin, 2012). Bakker (2006a) recommended that the skewness of a cell should be as follows:

   - Hexahedron and quadrilateral cells: skewness should not exceed 0.85
   - Triangle cells: skewness should not exceed 0.85
   - Tetrahedron cells: skewness should not exceed 0.9

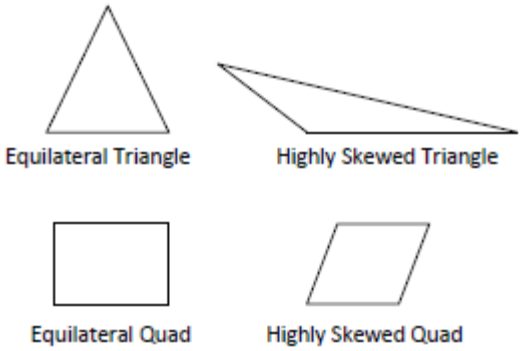


Figure 3.7: Ideal and skewed triangles and quadrilaterals (Asyikin, 2012)

2. Smoothness and aspect ratio

   Bakker (2006a) stated that the quality of the mesh can be assessed based on change of cells in size and the aspect ratio. The change of cells in size should be not too significant as illustrated in Figure 3.8a. The aspect ratio is the ratio of the longest edge length to shortest edge length. It is best to have an aspect ratio as close as possible to 1, since an aspect ratio equal to 1 means that the cell is equilateral (refer to Figure 3.8b). However, an aspect ratio may be neglected for a fully developed flow.

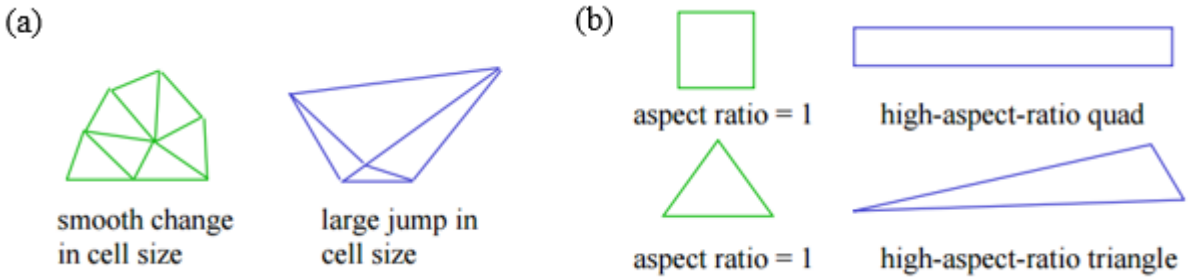


Figure 3.8: Examples of: (a) smooth and large change in cell size, (b) Ideal and high aspect ratio cells (Bakker, 2006a)

In addition, there are some guidelines that need to be followed while generating a mesh. Bakker (2006a) explained that those guidelines are as follows:

1. Resolution

The grid for pertinent flow features (e.g. fluid flow in near wall region where no-slip condition is assumed as the boundary condition of the wall) should be adequately sufficiently resolved, the examples mesh resolution in near wall regions are presented in Figure 3.9a. Additionally, a

quadrilateral or hexahedron cells can be stretched if the flow is fully developed and essentially one-dimensional (refer to Figure 3.9b)

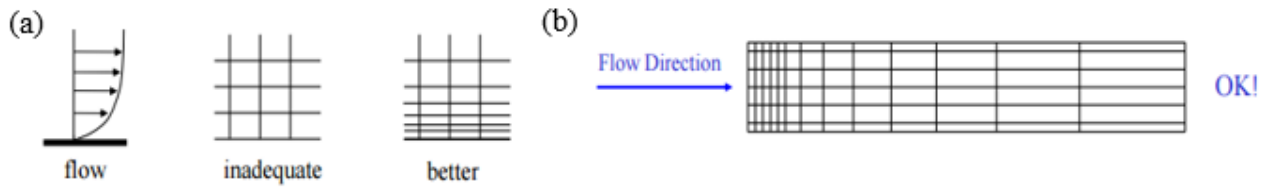


Figure 3.9: Grid examples for: (a) near wall regions and (b) fully developed flow region (Bakker, 2006a)

2. Smoothness

Mesh quality can be assessed based on its smoothness. Ideally, as suggested by Bakker (2006a), the maximum change in grid spacing should be less than 20%. The definition sketch of grid spacing is presented in Figure 3.10.

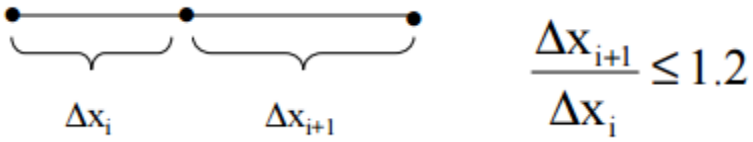


Figure 3.10: Illustration of grid spacing (Bakker, 2006a)

3. Total cell count

   More cells could give higher accuracy, but the drawback is it will require more memory and CPU time. Therefore, the cells number should be kept as sufficiently as possible, where the number is enough to give an accurate result and does not require too much memory. If the cells number should be decreased, it is suggested by Bakker (2006a) to use a non-uniform grid to cluster cells only where they are needed and to use solution adaptation to further refine only selected areas (refer to Figure 3.11 for an example of grid adaptation)

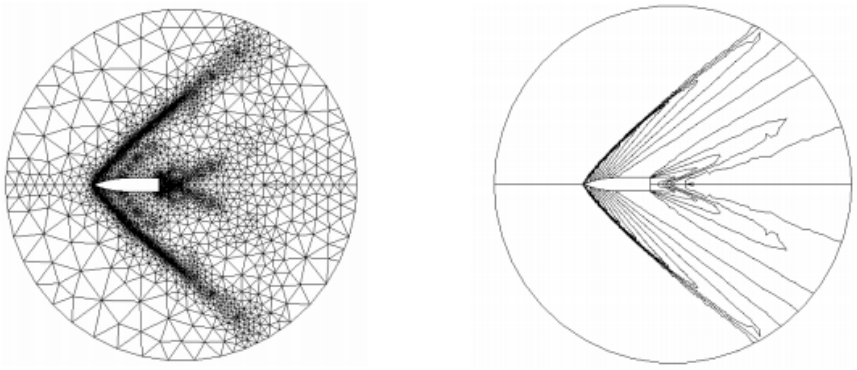


Figure 3.11: An example of grid adaptation (Bakker, 2006a)

In OpenFOAM, the information regarding mesh quality can be extracted by executing `checkMesh` utility in the command line.

3.6.3. Courant number

If stability problem occurs in OpenFOAM, small Courant numbers and relaxing the equation can help stabilizing the equation. Physically, the Courant numbers indicate how the fluid is moving through the cells. If the Courant numbers are less than 1, fluid particles move from one cell to another within one time step. If it is higher than 1, it means that the fluid particles move though two or more cells at each time step and it negatively affects the convergence. The Courant number is formulated as follows

For one-dimensional case: $$C = \frac{U.\Delta t}{\Delta x} \quad (57)$$

For two-dimensional case: $$C = \frac{U_x.\Delta t}{\Delta x} + \frac{U_y.\Delta t}{\Delta y} \quad (58)$$

Where U is the magnitude of the velocity, Δt is the time step, Δx and Δy are the length interval. Generally, it is always a good idea to look at the Courant numbers if we have convergence problems. It also indicates the accuracy of a transient simulation.

CHAPTER 4 - Fixed Pipe Cases Setup

4.1. Pre-processing

4.1.1. Mesh Generation

Based on the grid evaluations that were done by Asyikin (2012) and Patnana, Bharti, and Chhabra (2009), it is concluded that the best mesh structure to represent fluid flow over a cylinder is a rectangular domain with hexahedron (prism with quadrilateral base) cells. Inside the rectangular domain, there will be a cylinder placed symmetrically between the two horizontal plane walls with a distance of *H/2* from each plane. The cylinder is placed at a downstream distance of L_d from the right side and at an upstream distance of L_u from the left side. The schematic illustration of the domain is presented in Figure 4.1.

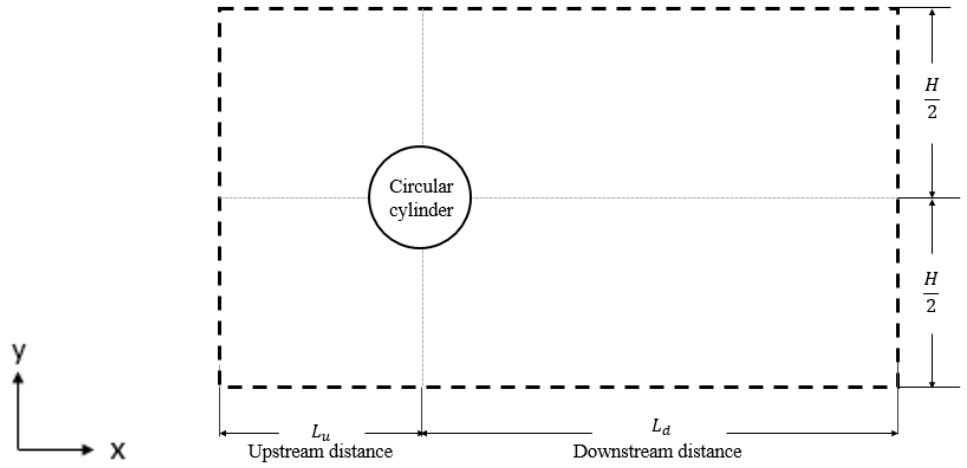


Figure 4.1: Domain illustration for flow over a cylinder

4.1.3.1. Mesh and Domain For Laminar Flow Case

The diameter (*D*) of the circular pipe in this thesis is 0.5 m. The pipe is assumed as a fixed circular cylinder, which means that it will not move even though it is subjected to forces, and its center is located in the coordinates (0, 0, 0). The mesh consists of 40 vertices and 12 blocks. The domain is made based on the schematic sketch in Figure 4.2 with L_u = 16D, L_s = 40D, and H = 40D. The vertices and blocks numbers are sorted based on the schematic sketch in Figure 4.3 and Figure 4.4. The details of `blockMeshDict` are presented in Appendix A.1.

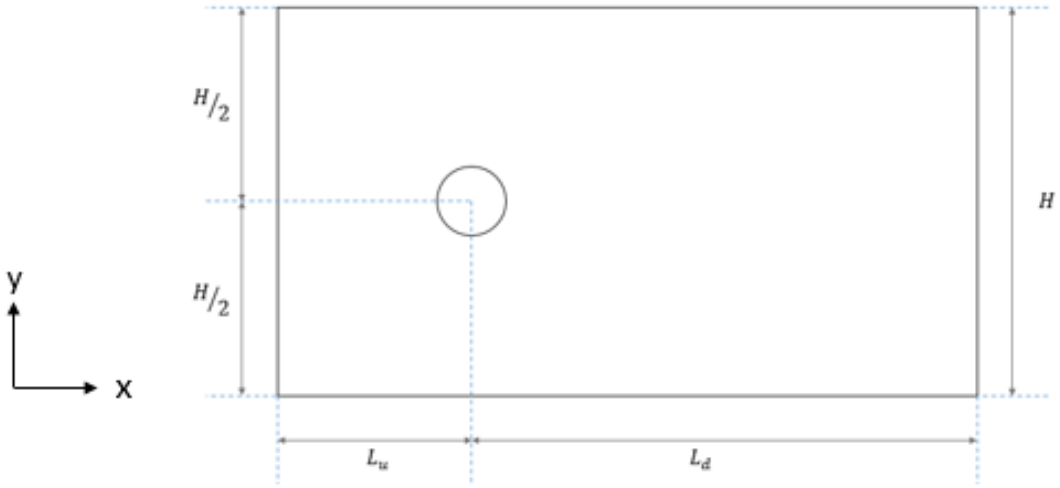


Figure 4.2: The schematic sketch of the domain for free fixed pipe

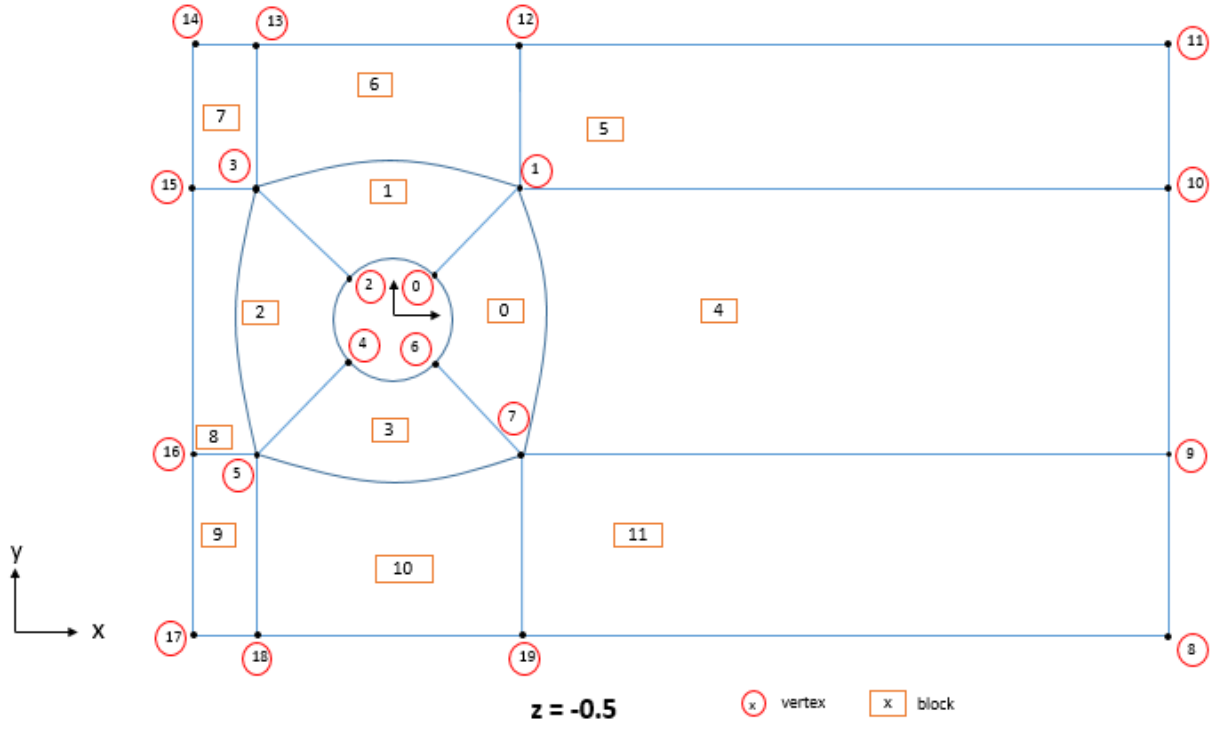


Figure 4.3: The location of vertices and blocks at plane z = -0.5

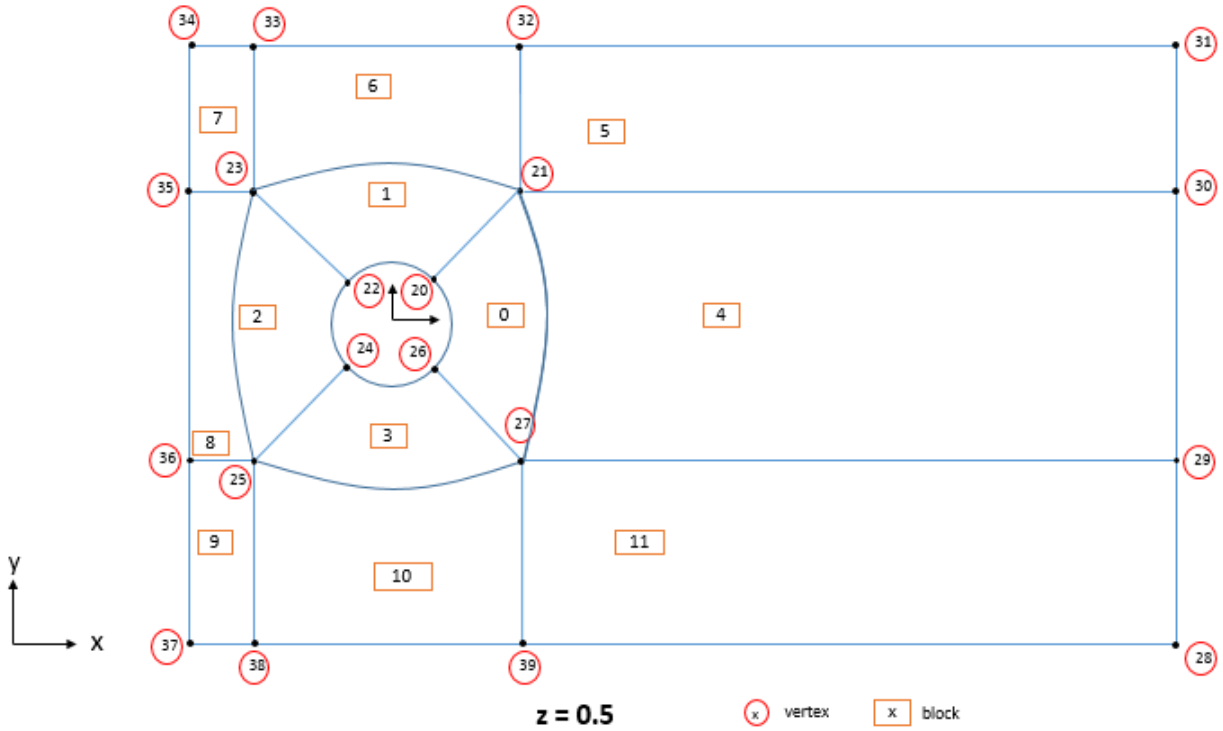


Figure 4.4: The location of vertices and blocks at plane z = 0.5

The mesh/grid is developed so that the aspect ratio will be higher near the domain boundaries. This arrangement is made since the flow is predicted to be a developed flow on that region (as explained before in section 3.6.2). The mesh around the wall, which in this case is the cylinder, should be finer so that it can generate a more accurate results of simulation. The mesh/grid near the cylinder is refined by modifying the grading scale in the *simpleGrading* under the `blocks` section.

The visualization of the grid/mesh in *paraView* is presented in Figure 4.5 while the detailed mesh visualization near the cylinder is shown in Figure 4.6. The parameters that determine the quality of the mesh can be observed by using *checkMesh* command in the terminal.

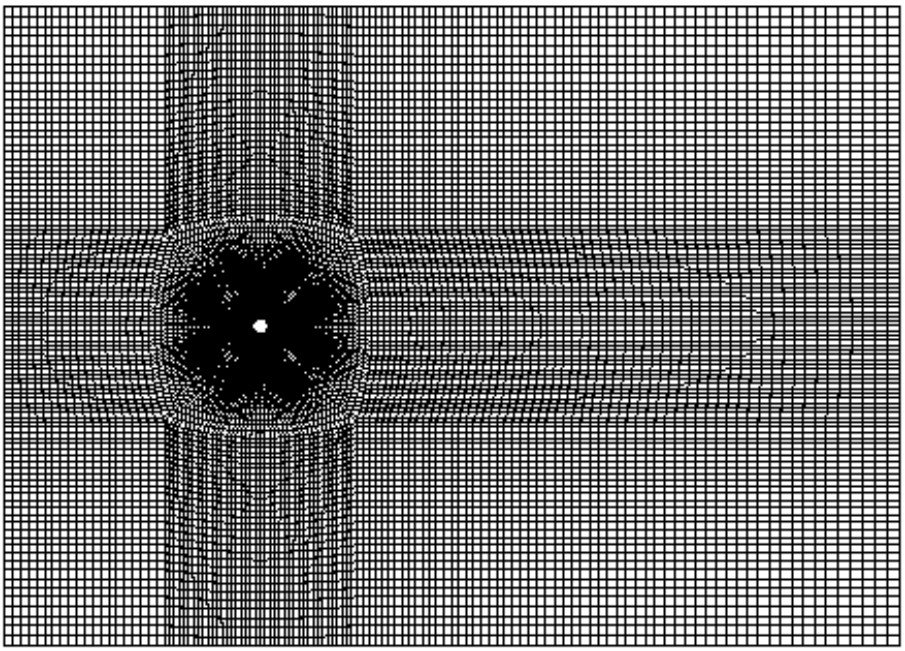


Figure 4.5: The visualization of the mesh in `paraView`

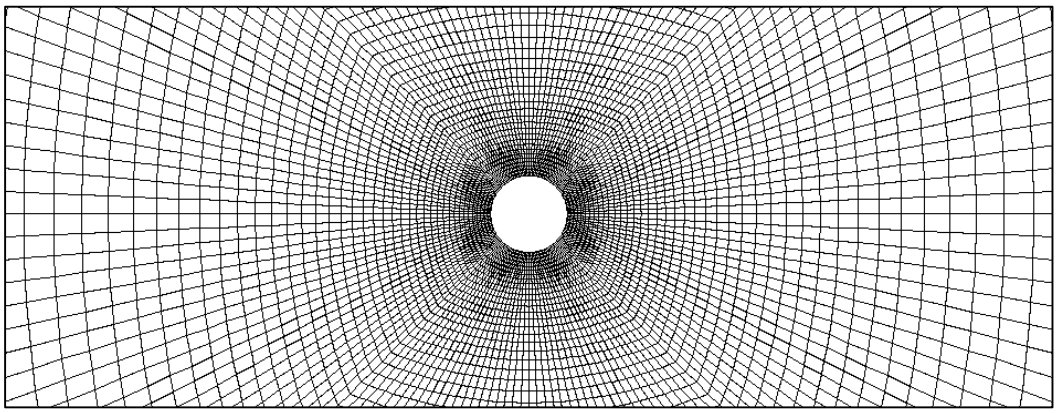


Figure 4.6: The visualization of the mesh near the cylinder in `paraView`

The result of *checkMesh* for the grid domain is shown in Figure 4.7. It can be seen that the skewness of the domain is equal to 0.611 and thus it is less than 0.85. This result agrees with the recommendation given by Bakker (2006a).

(a)

```
Mesh stats
    points:           34200
    internal points:  0
    faces:            67500
    internal faces:   33300
    cells:            16800
    faces per cell:   6
    boundary patches: 7
    point zones:      0
    face zones:       0
    cell zones:       0

Overall number of cells of each type:
    hexahedra:     16800
    prisms:        0
    wedges:        0
    pyramids:      0
    tet wedges:    0
    tetrahedra:    0
    polyhedra:     0
```

(b)

```
Checking geometry...
    Overall domain bounding box (-8 -10 -0.5) (20 10 0.5)
    Mesh (non-empty, non-wedge) directions (1 1 0)
    Mesh (non-empty) directions (1 1 0)
    All edges aligned with or perpendicular to non-empty directions.
    Boundary openness (1.33399e-18 -1.58482e-18 1.51476e-14) OK.
    Max cell openness = 2.20581e-16 OK.
    Max aspect ratio = 2.61679 OK.
    Minimum face area = 0.000211283. Maximum face area = 0.392519.  Face area magnitudes OK.
    Min volume = 0.000211283. Max volume = 0.126794.  Total volume = 559.804.  Cell volumes OK.
    Mesh non-orthogonality Max: 25.5243 average: 8.68134
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 0.611024 OK.
    Coupled point location match (average 0) OK.

Mesh OK.
```

Figure 4.7: The results of `checkMesh` showing parameters such as (a) types and number of cells, (b) aspect ratio, skewness, orthogonality, etc.

4.1.3.2. Mesh and Domain For Turbulent Flow Case

The schematic sketch for turbulent flow domain is similar with Figure 4.2, but with different length. The domain length for turbulent flow is set to be L_u = 8D, L_s = 40D, and H = 20D. The result of `checkMesh` for the grid domain is shown in Figure 4.8. It can be seen that the skewness of the domain is equal to 0.542 and thus it is less than 0.85. This result agrees with the recommendation given by Bakker (2006a).

(a)

```
Mesh stats
    points:           22812
    internal points:  0
    faces:            44886
    internal faces:   22074
    cells:            11160
    faces per cell:   6
    boundary patches: 7
    point zones:      0
    face zones:       0
    cell zones:       0

Overall number of cells of each type:
    hexahedra:     11160
    prisms:        0
    wedges:        0
    pyramids:      0
    tet wedges:    0
    tetrahedra:    0
    polyhedra:     0
```

(b)

```
Checking geometry...
    Overall domain bounding box (-4 -5 -0.5) (20 5 0.5)
    Mesh (non-empty, non-wedge) directions (1 1 0)
    Mesh (non-empty) directions (1 1 0)
    All edges aligned with or perpendicular to non-empty directions.
    Boundary openness (2.14796e-18 -3.51254e-18 -1.81129e-15) OK.
    Max cell openness = 2.20581e-16 OK.
    Max aspect ratio = 2.61679 OK.
    Minimum face area = 0.000211283. Maximum face area = 0.392519.  Face area magnitudes OK.
    Min volume = 0.000211283. Max volume = 0.0785038.  Total volume = 239.804.  Cell volumes OK.
    Mesh non-orthogonality Max: 25.5243 average: 10.2339
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 0.541629 OK.
    Coupled point location match (average 0) OK.

Mesh OK.
```

Figure 4.8: The results of `checkMesh` showing parameters such as (a) types and number of cells, (b) aspect ratio, skewness, orthogonality, etc.

4.1.3.3. Mesh and Domain For Pipe with Seabed Proximity Case

Pipe with proximity to seabed means that there is a certain distance between the cylinder and the seabed. This particular distance is often called as span height. The schematic sketch of pipe domain with different span height, *e*, is given in Figure 4.9a. The lengths of the domain are set as follows be $L_u = 8D$, $L_s = 40D$, and $H = 20D$. In this thesis, there will be a total of 4 cases with different gap ratio (*e/D*) which are 1, 0.5, 0.25, and 0. When the span height is equal to 0, it means that the pipe is located on the seabed as shown in Figure 4.9b.

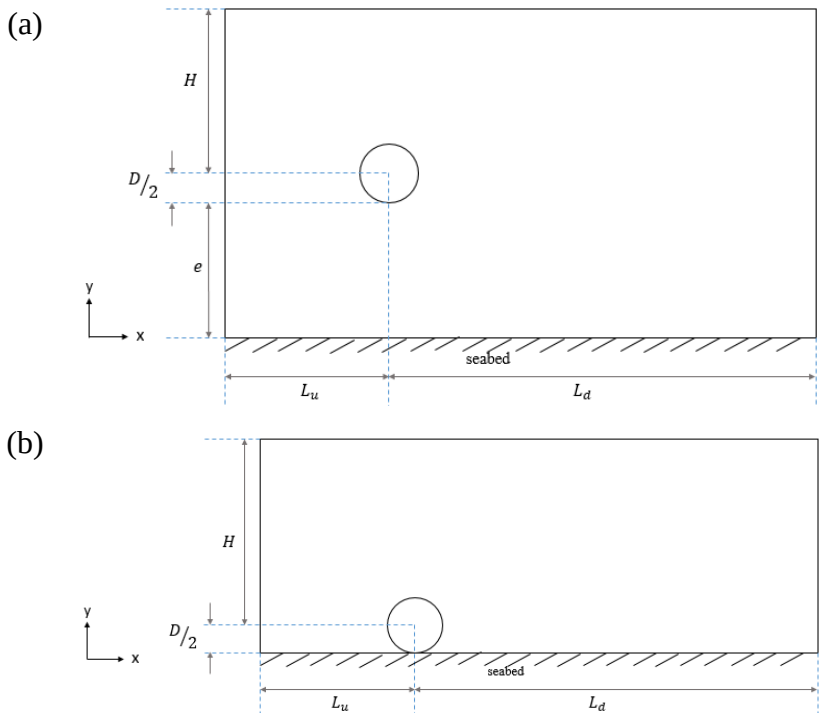


Figure 4.9: The schematic sketch of the domain for (a) pipe with proximity to seabed; (b) pipe on the seabed

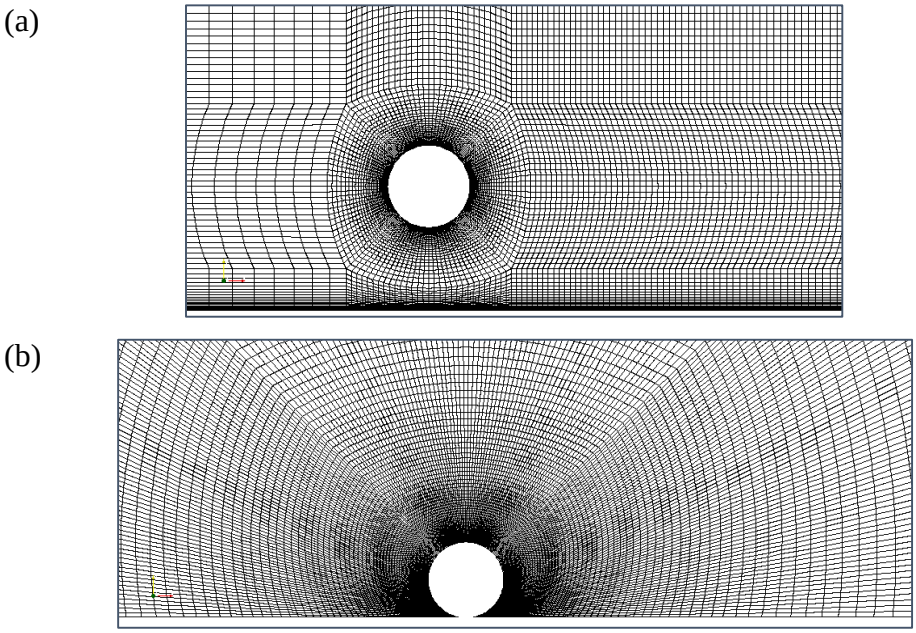


Figure 4.10: The visualization of the mesh for (a) pipe with proximity to seabed; (b) pipe on seabed

In the case of pipe with proximity to seabed, the mesh near the pipe and seabed is made to be very fine as seen in Figure 4.10a. The same condition applies in the case of pipe on the seabed, where the regions near the pipe and seabed is made to be finer than the other regions as seen in Figure 4.10b. Each case will have a different domain and the results of `checkMesh` for all domains are presented in Table 4.1. The skewness are mostly within the limit, which is 0.85, except for the pipe on the seabed mesh. The skewness for the mesh of pipe on the seabed is very high because the angle between the pipe and seabed is very small and this condition is unavoidable for a pipe located on the seabed.

Table 4.1: Mesh information for each case

Gap ratio (*e/D*)	Number of cells	Max skewness
1	49200	0.597
0.5	82650	0.539
0.25	45750	0.45
0	23750	1.036

4.1.2. Turbulence Model Selection

Since the k-epsilon model is not fit for unconfined flow and near wall region, the k-omega SST model is chosen for the turbulent flow simulation. This model has been popular because it can automatically transform into k-epsilon model in the developed flow and stays as k-omega model in near wall region. It was also mentioned by Zhao, Kaja, Xiang, and Yan (2013) that k-omega SST model gives a good prediction of the adverse pressure gradient flows.

4.1.3. Boundary Conditions Set Up

4.1.3.1. Laminar Flow

Assume two-dimensional, incompressible, uniform horizontal flow past over a fixed circular pipe (refer to Figure 4.11). The flow condition is made by considering the flow in a channel with the pipe placed symmetrically in between the two plane walls with slip boundary condition (refer to Figure 4.12)

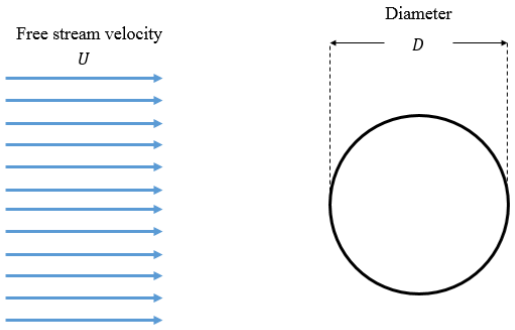


Figure 4.11: Illustration of flow over an unconfined cylinder

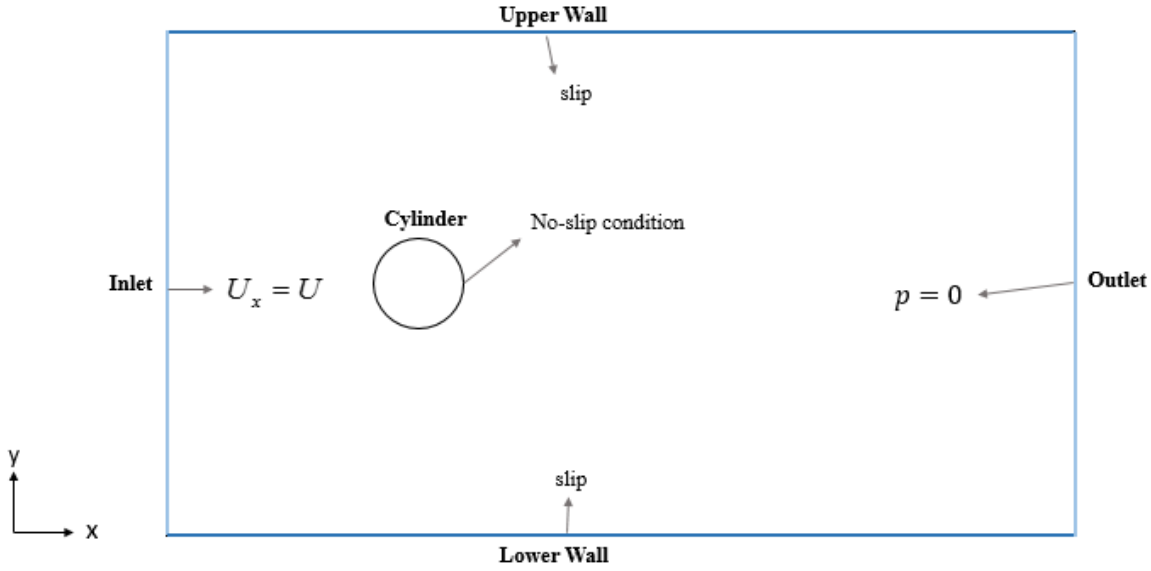


Figure 4.12: Schematic illustration for boundaries in laminar flow

The boundary conditions for the velocities in the unconfined flow are as follows:

- At the inlet: The uniform flow conditions set at the inflow are as follows

$$U_x = U, \qquad U_y = 0, \qquad \text{and} \qquad U_z = 0 \tag{59}$$

- At the outlet: A zero diffusion for all the flow variables, which means that the conditions on the outflow boundary are extrapolated from within the domain and resulted in negligible influence to the upstream flow conditions. The extrapolation updates the pressure and velocity in the outflow in a way that is consistent with the fully-developed flow assumption, which is no area change at the outflow boundary (Patnana et al., 2009).

$$\frac{\partial p}{\partial t} + \bar{U}_n \frac{\partial p}{\partial n} = 0 \quad (60)$$

- At the upper and lower wall: The slip condition, where viscous effects on the wall are negligible, is assumed for these boundaries.

- On the surface of the pipe: The no-slip condition assumption is used, whereas all velocity components on the pipe's wall will be equal to zero due to the viscous effects of the fluid

- The boundary condition of the front and back side of the domain is set as `empty` since the flow is intended to be simulated as a 2-dimensional flow.

In this thesis, there will be a total of 11 cases with different Reynolds number simulated as laminar flow. Realistically, the velocity should have been changed whenever the Reynolds number is changed. Table 4.2 summarizes the velocity of each case is laminar flow simulation.

Table 4.2: Velocity and transit time of the laminar flow simulations

Reynolds Number	U (m/s)	Transit Time (s)
20	4.02×10^{-5}	697211
100	2.01×10^{-4}	139442
200	4.02×10^{-4}	69721
300	6.02×10^{-4}	46481
400	8.03×10^{-4}	34861
500	1.00×10^{-3}	27888
600	1.20×10^{-3}	23240
700	1.41×10^{-3}	19920
800	1.61×10^{-3}	17430
900	1.81×10^{-3}	15494
1000	2.01×10^{-3}	13944

It can be seen in Table 4.2 that each simulation takes a very long transit time and thus it requires a lot of memory. Therefore, the inlet velocity in the laminar flow simulation will be assumed to always stay the same (U = 1 m/s) instead. This kind of setting is applied to reduce the transit time in OpenFOAM and it is acceptable for laminar flow only. In return of keeping the inlet velocity to stay the same, the kinematic viscosity, υ, will be changed in every case instead to modify the Reynolds number. The kinematic viscosity is modified according this following formula

$$\upsilon = \frac{U \cdot D}{\mathrm{Re}} \tag{61}$$

The value of the kinematic viscosity is modified through the `transportProperties` directory under the `constant` folder. The value of kinematic viscosity for each case is presented in Table 4.3. To get the time averaged parameters, the simulation time should be at least longer than the transit time. Whereas transit time is the time needed for the fluid to flow from the inlet to the outlet.

Table 4.3: List of kinematic viscosity values for all cases

Reynolds Number	Kinematic Viscosity ($m^2.s^{-2}$)
20	0.025
100	0.005
200	2.5×10^{-3}
300	1.667×10^{-3}
400	1.25×10^{-3}
500	1×10^{-3}
600	8.333×10^{-4}
700	7.143×10^{-4}
800	6.25×10^{-4}
900	5.556×10^{-4}
1000	5×10^{-4}

The transit time is calculated according to this following formula

$$\text{Transit time} = \frac{L_u + L_d}{U} = \frac{8+20}{1}\frac{m}{m/s} = 28s \tag{62}$$

In this thesis, it is decided that the simulation for laminar flow will be at least 2 times longer than transit time, which finally produced

Simulation time = *transit time x 2* = 58s ≈ 60s

The examples of time directories such as velocity and pressure are presented in Appendix B.1 and Appendix B.2 respectively.

4.1.3.2. Turbulent flow

In turbulent flow simulation, we are required to specify the values of the turbulence variables at the inlet. We are using `kOmegaSST` model to simulate to turbulent flow in this thesis, which means

that the turbulence variables that need to be specified are k and ω at the inlet. According to Saxena (2015), the values for k and ω at the inlet are specified by using these following formulas

$$k = \frac{3}{2}(I \cdot U)^2 \tag{63}$$

$$\omega = \frac{0.09 \cdot k}{\beta \cdot \upsilon} \tag{64}$$

For external flow, such as flow over pipe in this thesis, the value of turbulent intensity can be as low as 0.5% depending on the flow characteristic. As for the turbulent viscosity ratio, β, it should be in the range between 0.1 and 0.2 so that the free stream turbulent viscosity will be on the order of laminar viscosity.

We should note that it is really important to specify appropriate turbulent quantities at the free stream boundaries for external flows. According to Saxena (2015), if the values of turbulent quantities are unphysical, it could cause the solution to be unrealistic and lead to divergence. The overall boundary conditions for turbulent flow domain is summarized in Figure 4.13

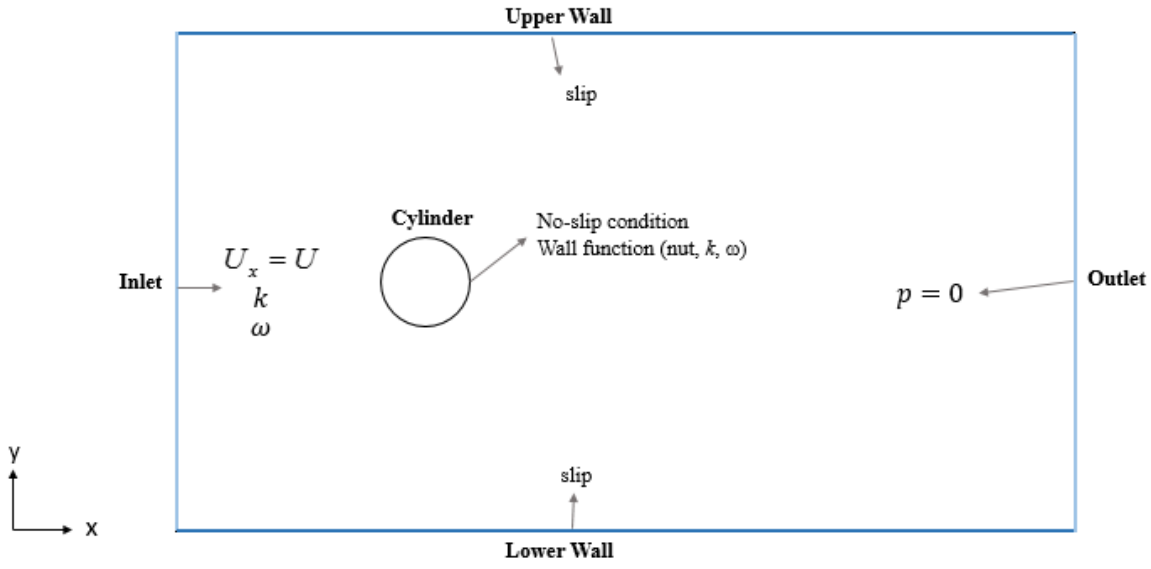


Figure 4.13: Schematic illustration of boundary conditions of turbulent flow

In this thesis, the turbulent intensity and turbulent viscosity ratio are specified to be 0.5% and 0.1 respectively. Since this is a turbulent flow simulation, we must use the real physical value for kinematic viscosity of water, υ, which is specified as 10^{-6} m^2s^{-1} in this thesis. The values of turbulent properties and free stream velocity are listed in Table 4.4

Table 4.4: Turbulent inlet properties and transit time

Reynolds number	U (m.s^{-1})	k (m^2.s^{-2})	ω (s^{-1})	Transit time (s)	Simulation time (s)
10,000	0.02	$1.5 \cdot 10^{-8}$	0.0135	1200	3000
100,000	0.2	$1.5 \cdot 10^{-6}$	1.3554	120	300
300,000	0.6	$1.36 \cdot 10^{-5}$	12.198	46.67	90
700,000	1.4	$7.409 \cdot 10^{-5}$	66.416	20	90
1,000,000	2	$1.5 \cdot 10^{-4}$	135.54	12	60
10,000,000	20.08	0.015	13554	1.394	10

The examples of time directories such as pressure, velocity, turbulent viscosity, intensity, and dissipation are presented in Appendix B.1 – B.5.

4.1.3.3. Pipe with seabed proximity

The simulation will be done for Re = 100, and hence the boundary conditions are similar to the boundary condition for Re = 100 in section 4.1.3.1. The only difference in this simulation is that the lower wall is now assumed to be representing a seabed. Therefore, the boundary condition of lower wall that was previously specified as slip in section 4.1.3.1 is now specified as no-slip condition. This kind of configuration is gained in consequence of assuming that the seabed is very rough which results in zero velocity on the surface of the seabed due to viscous effects of the fluid. The overall boundary conditions for pipe with proximity to seabed case is illustrated in Figure 4.14 and Figure 4.15.

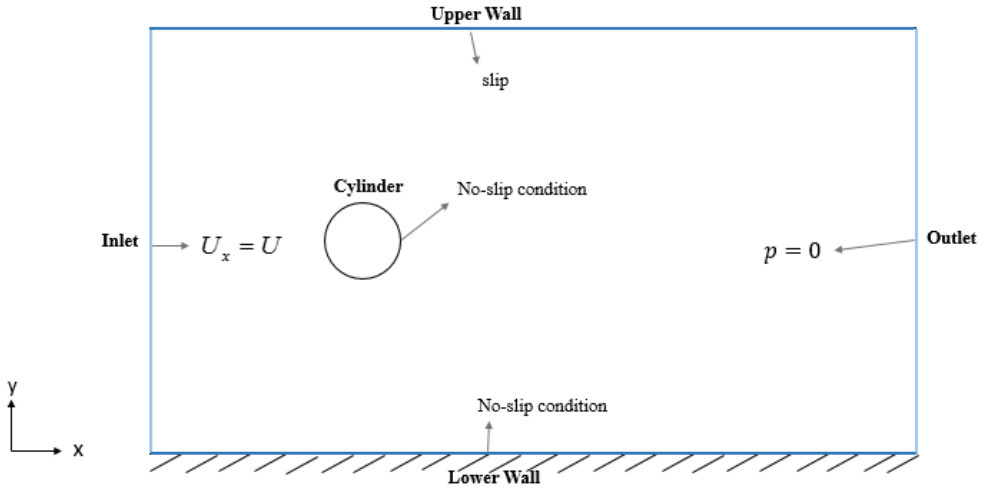


Figure 4.14: Schematic illustration of boundary conditions of pipe with proximity to seabed case

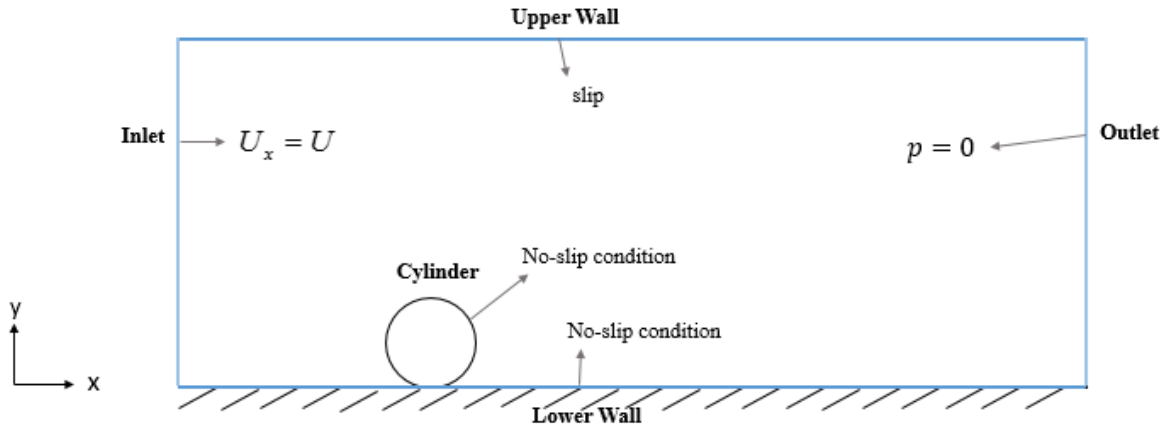


Figure 4.15: Schematic illustration of boundary conditions of pipe on the seabed case

4.2. Solutions Setup

4.2.1. Time and Data Input/Output Control

The OpenFOAM solvers begin all runs by setting up a database that controls the input/output (Greenshields, 2015). The `controlDict` dictionary sets input parameters that are essential for the creation of the database, i.e. time step and output time interval. The time step must be adjusted so that it reaches a low Courant number and gives accurate solution, especially for transient problems. When the velocity of the flow is increased, the time step of the simulation must be decreased so balance out the Courant number. In this thesis, the time step is modified so that the Courant number always stays below 0.2. Table 4.5 summarizes the essential parameters of the `controlDict` file of the simulations in this thesis.

Table 4.5: Summary of the time and data input/output control file

Reynolds Number	Time Step	Output Time Interval	Simulation Time
Laminar flow simulation			
Re ≤ 1000	0.002	0.05	60
Turbulent flow simulation			
10,000	0.05	1	3000
100,000	0.005	0.05	300
300,000	0.001	0.05	90
700,000	0.0005	0.05	90
1,000,000	0.0005	0.05	30
10,000,000	0.0001	0.05	10
Pipe with seabed proximity			
400	0.002	0.05	60

Additional functions can also be added to the `controlDict` to write down some other information from the simulation. For example, this thesis added additional function, `forceCoeffs`, to extract the force coefficients from the simulation. The `controlDict` directories for the laminar and turbulent flow are presented in Appendix C.1 and Appendix C.2 respectively.

4.2.2. Discretization Scheme

According to B. H. Hjertager (2009), discretization is essential so that all equation terms can be solved. There are many options for the discretization schemes available in OpenFOAM and all are well maintained. The discretization schemes for terms that appear in the running application are set by the `fvSchemes` dictionary in the `system` directory. The `fvSchemes` is made of several sub-dictionaries for different types of terms to be discretized. According to B. H. Hjertager (2009), those sub-dictionaries are as follows:

- `fvSchemes/ddtSchemes`: specifies the first time derivative of the terms. In this thesis, the time derivative scheme is set as `backward`, which is a genuine, transient, and implicit schemes that has 2nd order accuracy.
- `fvSchemes/divSchemes`: contains the divergence terms and probably is the most important discretization scheme in CFD. The method of this discretization scheme is always using `Gauss` scheme and it requires a selection of the interpolation scheme for the dependent field. In this thesis, the interpolation schemes are set as `linear` for the flux and `limitedLinear` for the convection terms. Both schemes have second order accuracy, but `linear` is unbounded while `limitedLinear` is bounded.
- `fvSchemes/gradSchemes`: computes gradients at cell centers. In this thesis, the discretization method is set as *Gauss*. The interpolation scheme is employed as *linear*.
- `fvSchemes/snGradSchemes`: discretizes the surface normal gradient terms. It is evaluated at a cell face; it is the component, normal to the face, of the gradient of values at the centers of the 2 cells that the face connects (Greenshields, 2015). In this thesis, the scheme is set as `limited` with a grade of correction of 0.5.
- `fvSchemes/laplacianSchemes`: discretizes the Laplacian terms. Interpolation scheme should be defined for diffusion coefficients. In this thesis, the discretization method for all

variables is `Gauss`, the interpolation scheme is `linear`, and the surface normal gradient scheme is set as `limited` with a grade of correction of 0.5.

- `fvSchemes/interpolationSchemes`: defines the schemes that are used to interpolate the variable at cell faces. In this thesis, the interpolation method for the velocity is set as `linear` which has 2nd order accuracy and unbounded
- `fvSchemes/fluxRequired`: defines the fields fir which the flux is generated in the application. In this thesis, the flux is generated from pressure since the fluxes are computed after solving the pressure equation (B. H. Hjertager, 2009).

The `fvSchemes` dictionaries of this thesis are presented in Appendix D.1 for the laminar flow and Appendix D.2 for the turbulent flow.

4.2.3. Solution and Algorithm Control

The equation solvers, tolerances, and algorithms are controlled from the `fvSolution` dictionary in the `system` directory. The solver that is used in the `fvSolution` directory of this thesis is `CAMG` (Geometric-agglomerated Algebraic Multigrid) which requires positive definite, diagonally dominant matrix. For `CAMG`, smoothing is a part of the multigrid method. It can reduce high frequency error on current mesh as it solves for symmetric and asymmetric matrices (B. H. Hjertager, 2009). The smoothing scheme that is applied in the solver is `GaussSeidel` which means that the smoothing step is based on the Gauss Seidel method. As explained by B. H. Hjertager (2009), `GAMG` for pressure and `smoothSolver` for the rest of the variable are recommended to be applied because of their speed.

The pressure-velocity coupling methods in laminar and turbulent cases are `SIMPLE` and `PISO` respectively. The `fvSolution` directories for laminar and turbulent flow simulation are presented in Appendix E.1 and Appendix E.2 respectively.

4.2.4. Solver

- Laminar Flow

The solver that is chosen for laminar flow simulation is `icoFoam`, which is a transient solver for incompressible, laminar flow of Newtonian fluids, The OpenFOAM Foundation (n.d.). The `icoFoam` solves the incompressible laminar Navier-Stokes equations using the PISO algorithm.

The code is inherently transient, requiring an initial condition and boundary conditions, (OpenFOAMWiki, 2009).

- Turbulent Flow

The solver that is chosen for turbulent flow simulation is `pisoFoam`, which is a transient solver for incompressible flow, The OpenFOAM Foundation (n.d.). A transient solver runs with time steps, where in each time step the fluctuations associated with the turbulence are captured (Douglas, 2014).

- Pipe with Proximity to Seabed

Since this case is simulated for laminar flow, the solver that is used for this case is `icoFoam`.

4.3. Post-processing

The desired output variables of the simulation will be the force coefficients, pressure, and velocity time histories. Variables such as pressure and velocity will be extracted from the simulation by using a utility called `probeLocations`. Several probes will be located in parallel to form a straight line in y = 0, as shown in Figure 4.16. The locations and the selection of the parameters that is going to be extracted (i.e. velocity, pressure, etc.) could be modified anytime by using the `probesDict` that is located under the `system` folder.

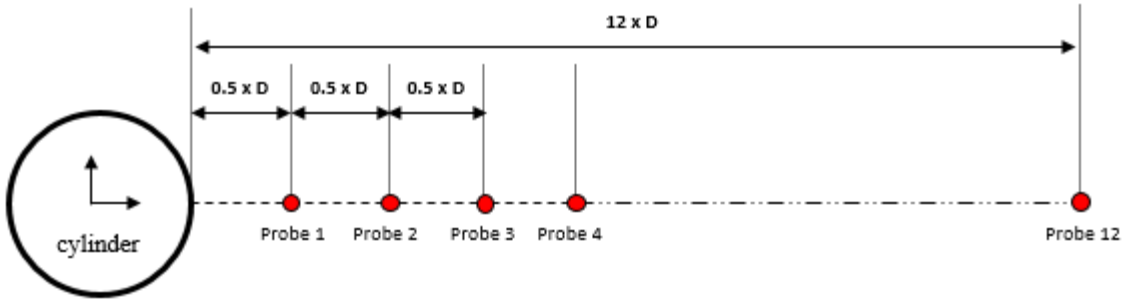


Figure 4.16: Locations of probes

The force coefficients such as drag and lift coefficients on the cylinder could be extracted by adding some lines in `controlDict` dictionary under keyword `forceCoeffs`. The essential information such as free stream velocity, diameter, reference area, etc. should be provided to generate the correct force coefficients. The visualization of the simulation results (i.e. velocity, pressure, viscosity, etc) can be seen by using the `paraView` utility.

CHAPTER 5 - Results and Discussions of Fixed Pipe Cases

5.1. Laminar Flow Simulation Results and Discussions

5.1.1. Results and Discussions of Laminar Flow with Re = 20

The visualization of velocity and pressure profiles for laminar flow with Re = 20 are shown in Figure 5.1 and Figure 5.2 respectively.

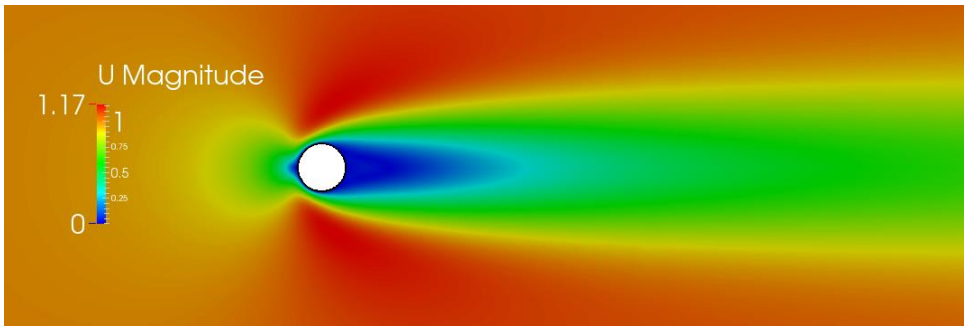


Figure 5.1: Horizontal velocity profile for laminar flow with Re = 20 in paraView

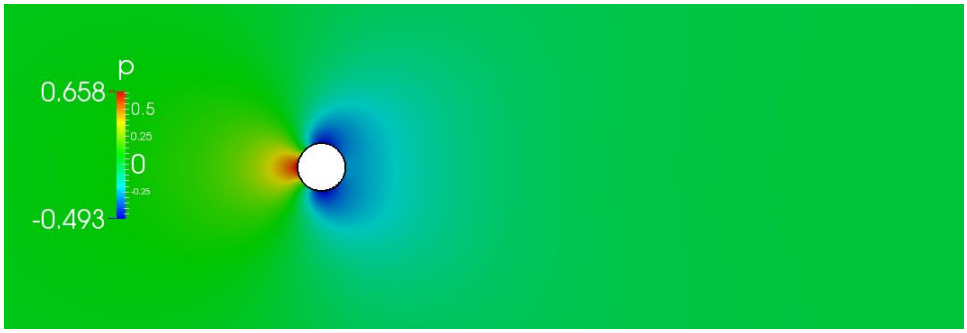


Figure 5.2: Pressure profile for laminar flow with Re = 20 in paraView

Based on the Figure 5.1 and Figure 5.2, the wake behind the cylinder always stays symmetrical along the simulation time and there is no vortex shedding sighted. This remark agrees with the expected result, as it was mentioned earlier in section 2.3 that vortex shedding phenomenon only occurs for flow with Reynolds number greater than 40.

Afterwards, the velocity and pressure measurements were generated from the probes location that were explained before in section 4.3. The velocity and pressure time histories are presented in

Figure 5.3 and Figure 5.4 respectively. The lift and drag coefficient time histories of the fixed pipe is presented in Figure 5.5.

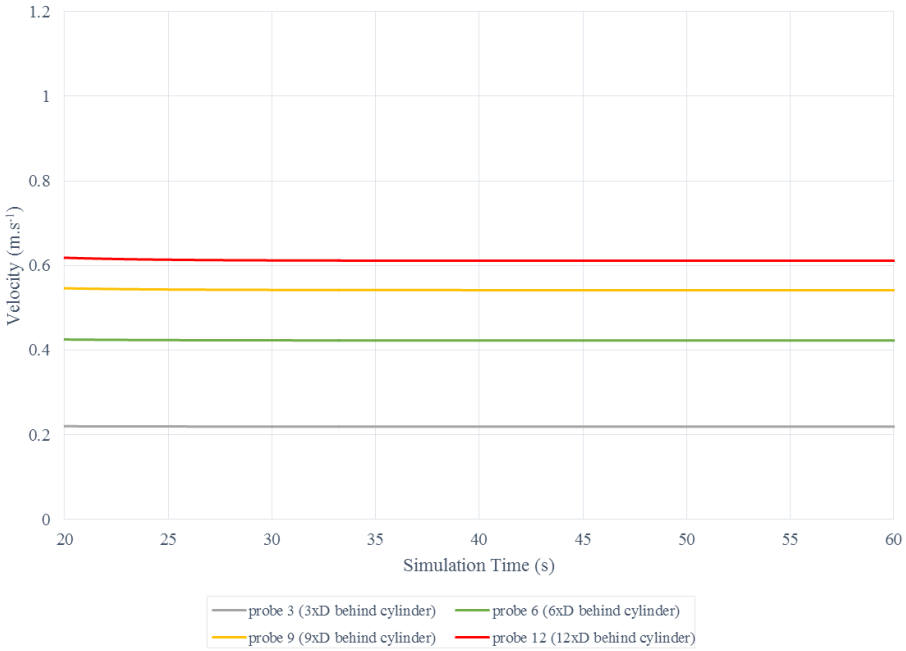


Figure 5.3: Horizontal velocity time histories for laminar flow with Re = 20

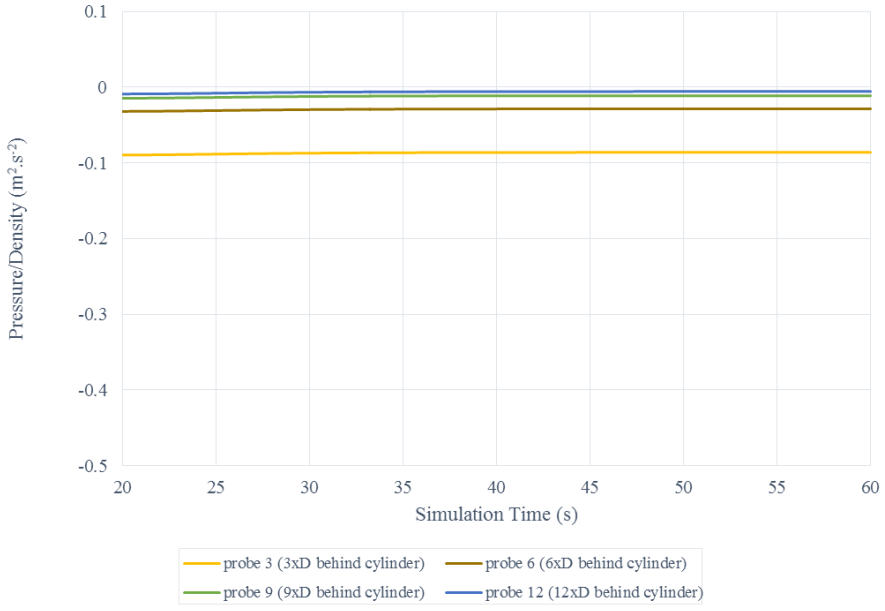


Figure 5.4: Pressure time histories for laminar flow with Re = 20

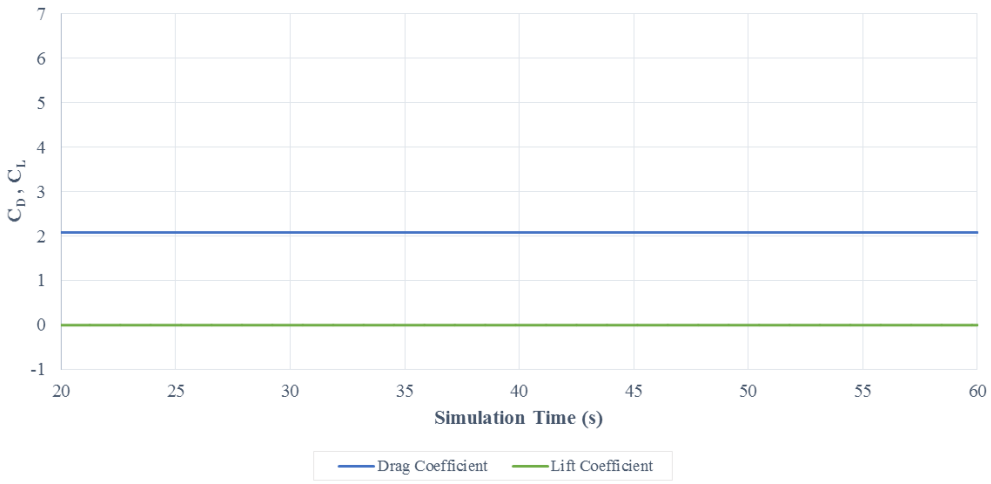


Figure 5.5: Hydrodynamic coefficients time histories for laminar flow with Re = 20

Additionally, based on the hydrodynamic coefficient time histories Figure 5.5, it can be seen that there is no value at all for lift coefficient and we get mean drag coefficient equal to 2.08. Since there is no vortex shedding behind the cylinder at all, it results in no pressure variation in the cross-flow direction. On the other hand, the drag coefficient is a non-zero because there is a wake created behind the cylinder which results in pressure variations in the in-line direction. However, the drag coefficient does not oscillate because no vortex is shed.

5.1.2. Results and Discussions of Laminar Flow with Re = 100

The visualization of velocity and pressure profiles for laminar flow with Re = 100 are shown in Figure 5.6 and Figure 5.7 respectively.

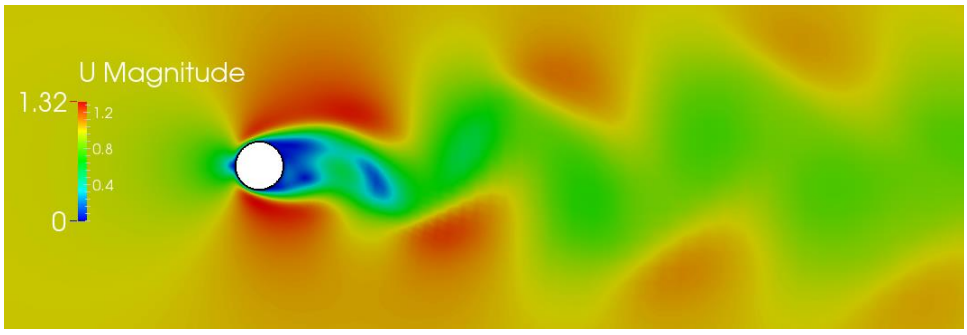


Figure 5.6: Horizontal velocity profile for laminar flow with Re = 100 in *paraView*

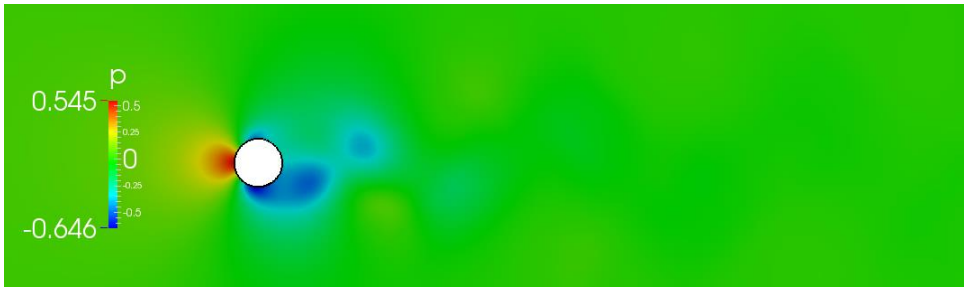


Figure 5.7: Pressure profile for laminar flow with Re = 100 in *paraView*

It can be seen from the Figure 5.6 and Figure 5.7 that the vortex shedding phenomena has occurred. This results compromise with the theories mentioned in section 2.3, which stated that the vortex street will start to appear for flow with Reynolds number greater than 40. In this range, the wake becomes unstable and eventually will result in vortex shedding that oscillates at a certain frequency.

The frequency of vortex shedding, f_v, can be measured from the velocity, pressure, or the lift coefficient time histories (as described previously in Figure 2.6). We can also measure the vortex shedding frequency based on the drag coefficient time histories but we should note that the period of the drag coefficient is only half of the vortex shedding period.

The velocity and pressure time series are also generated in the probes that were explained before in section 4.3. The velocity and pressure time histories for flow with Re = 100 are presented in Figure 5.8 and Figure 5.9 respectively.

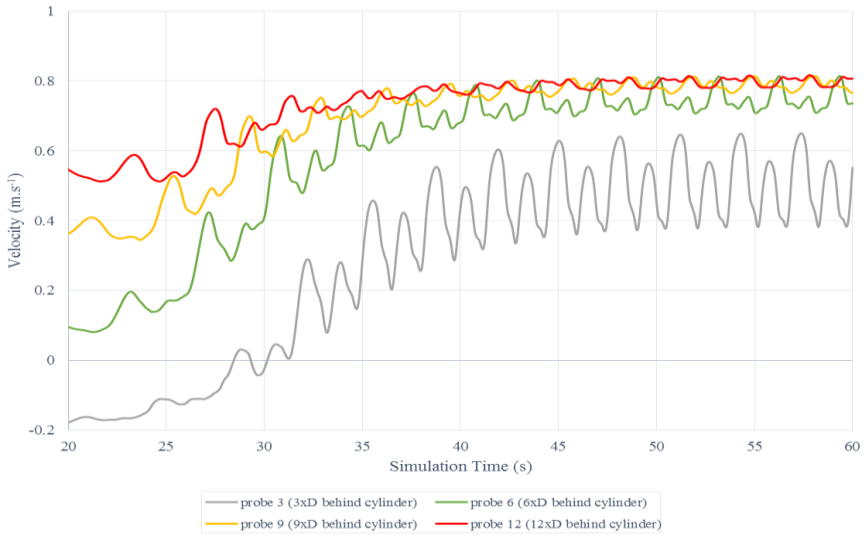


Figure 5.8: Velocity time histories for laminar flow with Re = 100

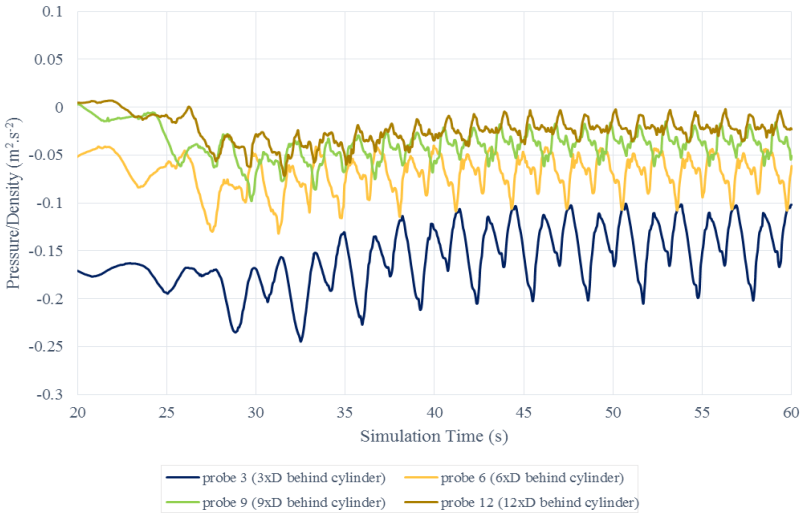


Figure 5.9: Pressure time histories for laminar flow with Re = 100

For now, we will try to measure the vortex shedding frequency from the velocity time series that were generated from each probe. After we have all the 12 different frequencies, we can proceed with the Strouhal number calculation using equation (4). Afterwards, the average Strouhal number can be calculated by using this following formula

$$St_{average} = \frac{\sum_{i=1}^{n} St_i}{n} \tag{65}$$

Where i indicates the number of probes and n indicates the total number of the probes, which in this case is 12. Finally we got an average vortex shedding frequency and Strouhal number equal to 0.326 Hx and 0.163 respectively as shown in Figure 5.10 and Figure 5.11.

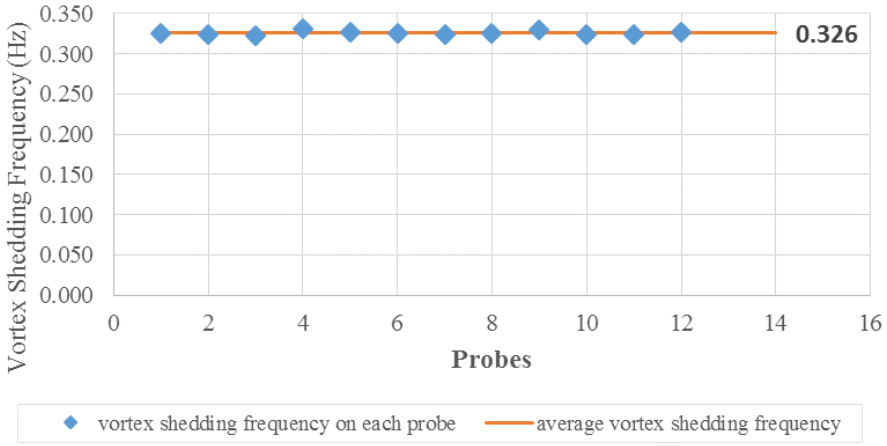


Figure 5.10: Vortex shedding frequency of laminar flow with Re = 100

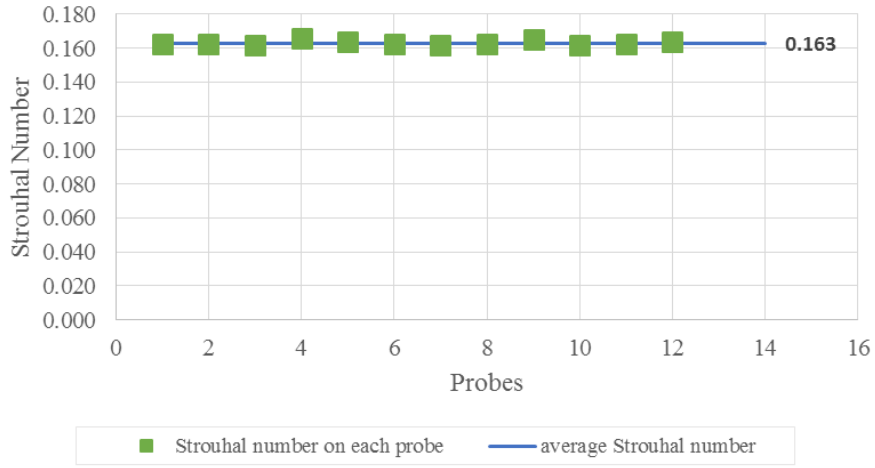


Figure 5.11: Strouhal number measurement of laminar flow with Re = 100

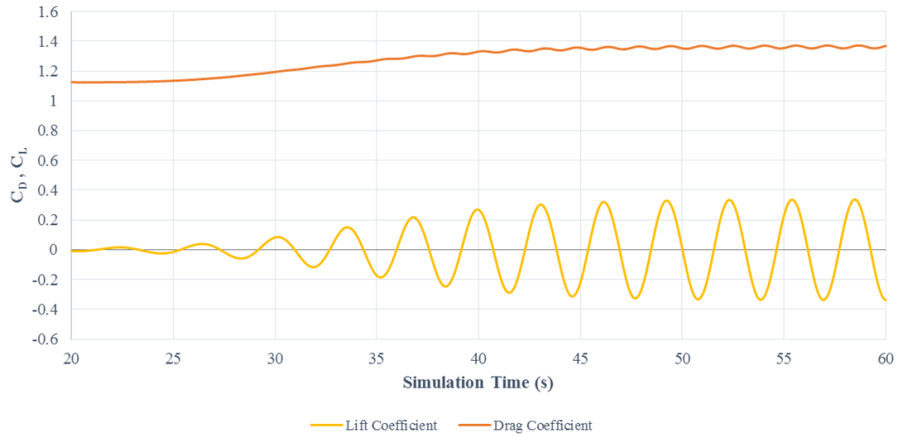


Figure 5.12 Hydrodynamic coefficients time histories for laminar flow with Re = 100

As mentioned earlier, another way to measure the vortex shedding frequency is by observing the oscillations of lift coefficient time histories, which is shown in Figure 5.12. From the figure, a Strouhal number of 0.1624 is obtained, which is very close to the Strouhal number that is obtained by observing the velocity time histories in Figure 5.11Figure 5.11 Approximately, the deviation between these Strouhal numbers is 0.33%. Therefore, we can conclude that the velocity, pressure, and lift coefficient are oscillating at the same frequency. The mean drag coefficient in this simulation is 1.361, and the amplitude of the lift coefficient is 0.338.

5.1.3. Results Summary and Discussions of Laminar Flow with $100 \leq Re \leq 1000$

5.1.3.1. Pressure and Velocity

The results for the rest of the laminar flow simulations are presented in Appendix G. It can be seen from the Figure 5.13 and Figure 5.14 that the mean velocity and its fluctuation increases for higher Reynolds number. In contrast, the mean pressure decreases for higher Reynolds number. This result is expected as we know that velocity is always inversely proportional to pressure according to the Bernoulli equation.

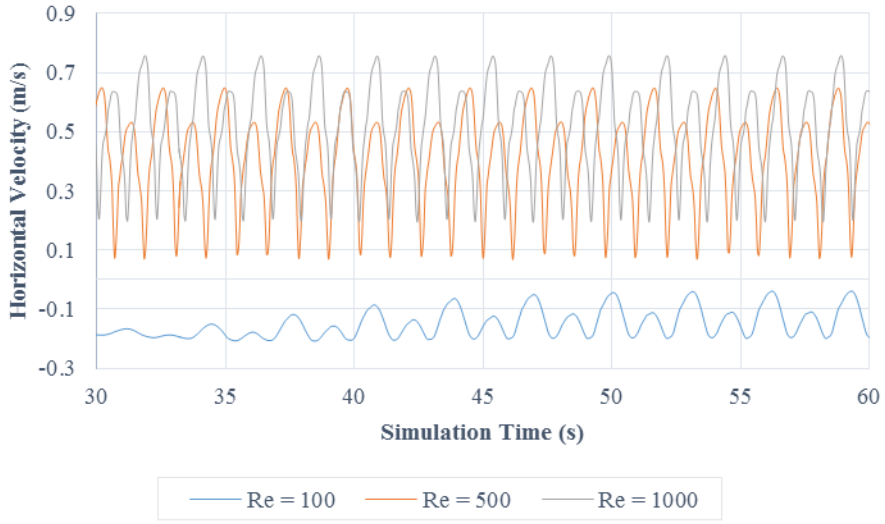


Figure 5.13: Horizontal velocity time histories of laminar flow at probe 1

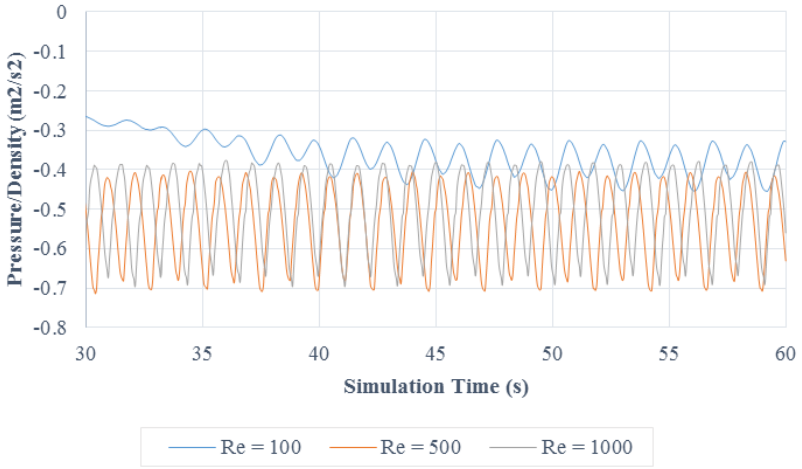
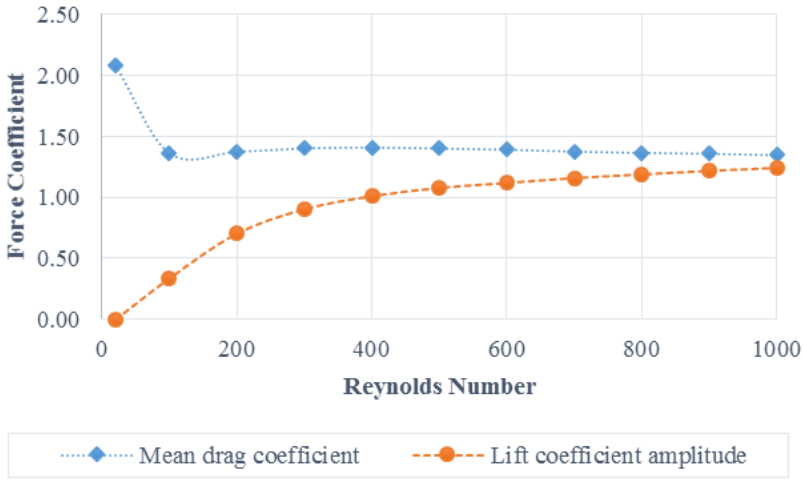


Figure 5.14: Pressure time histories of laminar flow at probe 1

5.1.3.2. Hydrodynamic Coefficients

The time series for drag coefficient and lift coefficient are presented in Figure 5.15 and Figure 5.16 respectively. The mean drag coefficient and lift coefficient amplitude are summarized in Table 5.1. The relationships between the Reynolds numbers and the hydrodynamic coefficients



are plotted in Figure 5.17.

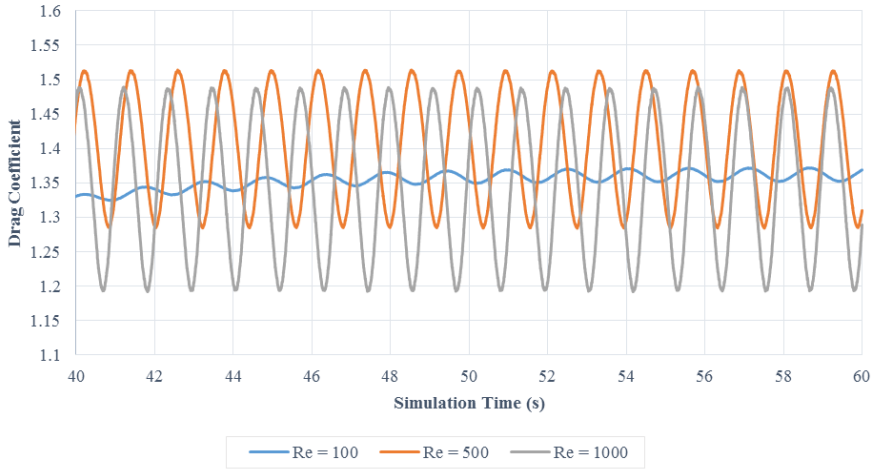


Figure 5.15: Drag coefficient time histories of laminar flow

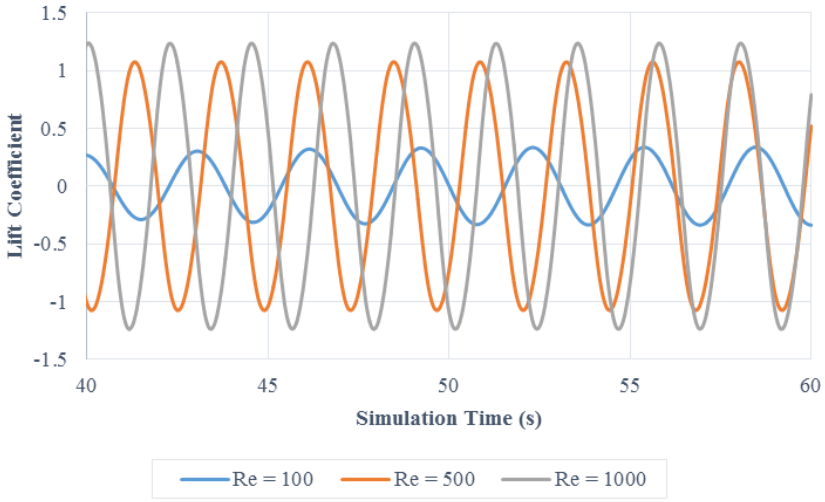


Figure 5.16: Lift coefficient time histories of laminar flow

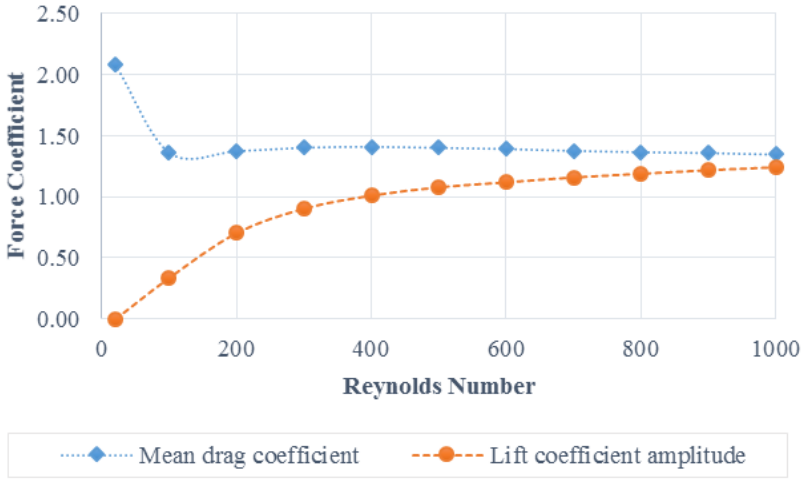


Figure 5.17: Force coefficient - Reynolds number relationship

Table 5.1: Hydrodynamic coefficients of laminar flow summary

Re	$\bar{C}_D$			C_L
20	2.081	±	0.00	0
100	1.361	±	0.01	0.338
200	1.374	±	0.05	0.703
300	1.403	±	0.08	0.902
400	1.410	±	0.10	1.009
500	1.401	±	0.11	1.076
600	1.392	±	0.12	1.117

Re	$\bar{C}_D$			C_L
700	1.377	±	0.13	1.157
800	1.366	±	0.14	1.187
900	1.359	±	0.14	1.216
1000	1.347	±	0.15	1.239

For the mean drag coefficient, there is a quite significant change when the Reynolds number shifts from 20 to 100. As the first vortex shedding phenomenon occurs in Re = 100, its occurrence probably has significantly decreased the mean drag forces. Afterwards, the mean drag coefficient appears to be stagnant as the Reynolds number increases. Additionally, we can see that the amplitude of lift coefficient increases exponentially as the Reynolds number increases.

Many studies have been done to measure the lift and mean drag coefficient for laminar flow over a cylinder, Table 5.2 presents the hydrodynamic coefficients gained from the past studies with the present study. It can be seen that the present study generates force coefficients that have quite a good agreement with force coefficients obtained it the other past studies.

Table 5.2: Comparison of force coefficients from the simulation results with the past studies

Source	Re = 100		Re = 200		Re = 1000	
	$\bar{C}_D$	C_L	$\bar{C}_D$	C_L	$\bar{C}_D$	C_L
Behr, Hastreiter, Mittal, and Tezduyar (1995)	1.37	0.371	-	-	-	-
Mittal and Raghuvanshi (2001)	1.402	0.355	-	-	-	-
Berthelsen and Faltinsen (2008)	1.38	0.34	1.37	0.7	-	-
Franke, Rodi, and Schönung (1990)	-	-	1.31	0.65	1.47	1.36
Present study	1.361	0.338	1.374	0.703	1.347	1.239

5.1.3.3. Strouhal number

The next parameter that we are going to observe is the Strouhal number. For each Reynolds number where vortex shedding is sighted, we can calculate the Strouhal number by measuring the vortex shedding period first (the vortex shedding period and frequency were explained in section 2.3 and 2.4).

Williamson and Brown (1998) had formulated a functional relationship between the Strouhal number and Reynolds number for the cylinder wake. The formulated functional relationship was

tested with the recent extension of the "numerical" laminar regime by Henderson (1997) for Reynolds number up to 1000 and it gave an excellent representation. The formula is written as an expansion and involves a truncated series of three terms as follows

$$St = 0.2731 - \frac{1.1129}{\sqrt{\mathrm{Re}}} + \frac{0.4821}{\mathrm{Re}} \tag{66}$$

Figure 5.18 shows the graph describing the relationship between the Reynolds number and Strouhal number, while Table 5.3 summarizes the measured vortex shedding frequency and the comparison between the Strouhal numbers obtained from the simulation with the one that is calculated by using the formula from Williamson and Brown (1998).

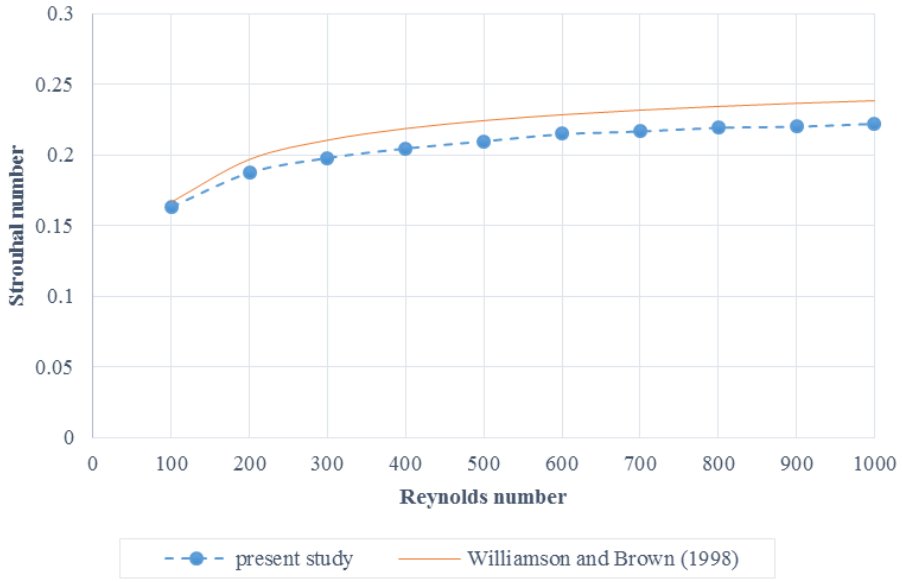


Figure 5.18 Strouhal number - Reynolds number relationship for Re ≤ 1000

Table 5.3: Comparison of the Strouhal numbers from the present studies and experiments

Re	f_v	St		
		present study	Williamson and Brown (1998)	deviation
100	0.326	0.163	0.167	2.21%
200	0.375	0.187	0.197	4.76%
300	0.396	0.198	0.210	5.98%
400	0.409	0.205	0.219	6.39%
500	0.420	0.210	0.224	6.48%
600	0.430	0.215	0.228	5.97%
700	0.434	0.217	0.232	6.45%

Re	f_v	St		
		present study	Williamson and Brown (1998)	deviation
800	0.439	0.219	0.234	6.36%
900	0.440	0.220	0.237	6.90%
1000	0.444	0.222	0.238	6.77%

We can conclude from the Table 5.3 that the deviation is always less than 10%, which means that there is a good agreement between the two sets of value.

5.2. Turbulent Flow Simulation Results and Discussions

5.2.1. Results and Discussion of Turbulent Flow with Re = 10,000

The visualization of turbulent flow properties in *paraView* are presented in Figure 5.19 to Figure 5.22.

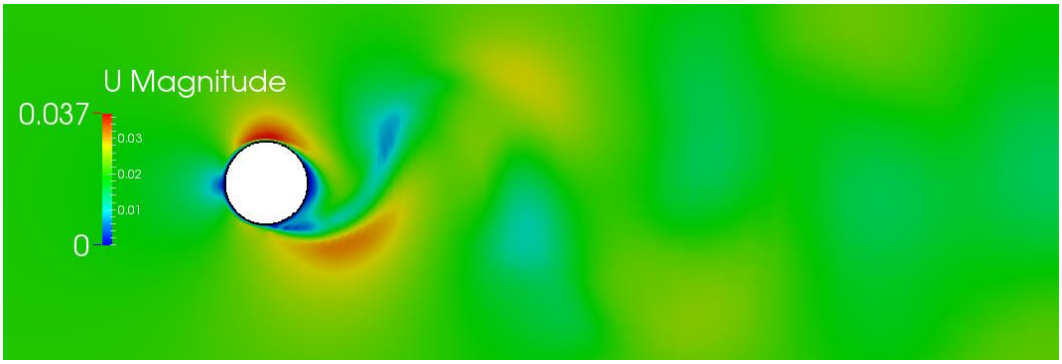


Figure 5.19: Velocity profile of turbulent flow with Re = 10,000 in paraView

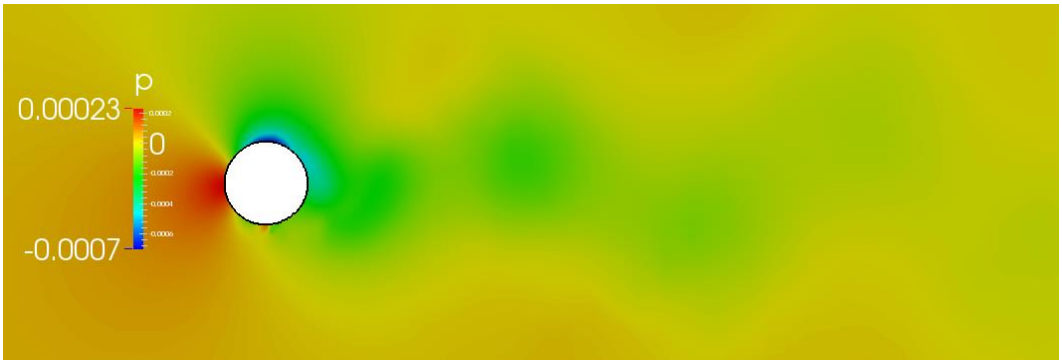


Figure 5.20: Pressure profile of turbulent flow with Re = 10,000 in paraView

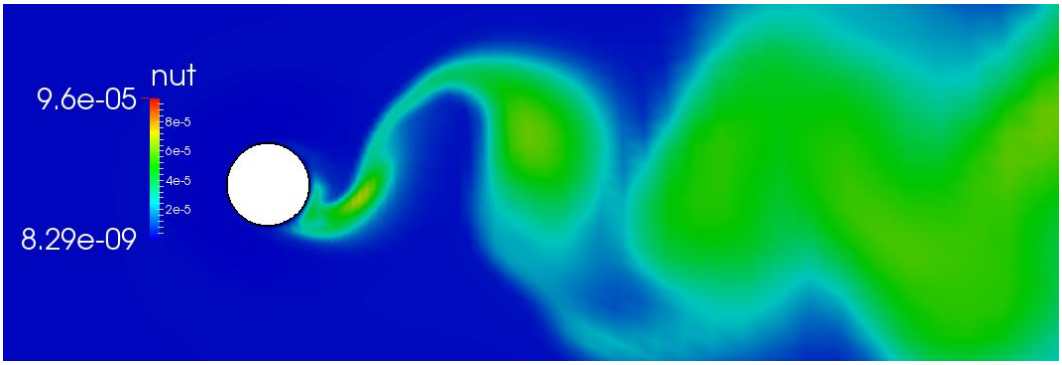


Figure 5.21: Turbulent viscosity profile of turbulent flow with Re = 10,000 in paraView

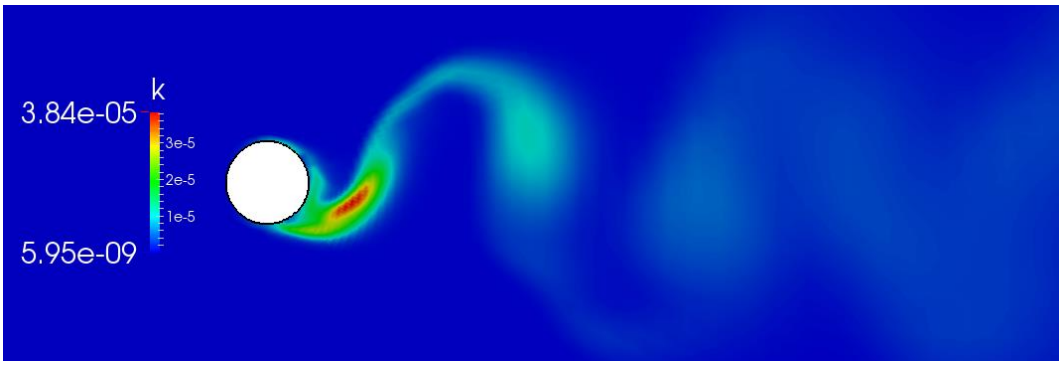


Figure 5.22: Turbulent kinetic energy profile of turbulent flow with Re = 10,000 in paraView

The wake behind the cylinder is oscillating due to the shear effect from the pipe wall. This oscillation implies an anisotropic structure of the turbulence. However, it can be seen in the Figure 5.23a that the oscillation is fading out in the region near the outlet and hence the fluctuations of the pressure (Figure 5.24) and velocity (Figure 5.25) is very small in that region. The main principle in the RANS model is that the turbulent viscosity is assumed to be isotropic. Therefore, when the flow is getting further from the cylinder, the turbulence properties will become more isotropic due to the absence of shear and then the turbulence cannot be sustained in the region any longer. Here in Figure 5.23b, the turbulent viscosity is very high near the outlet due to the isotropic properties of RANS model.

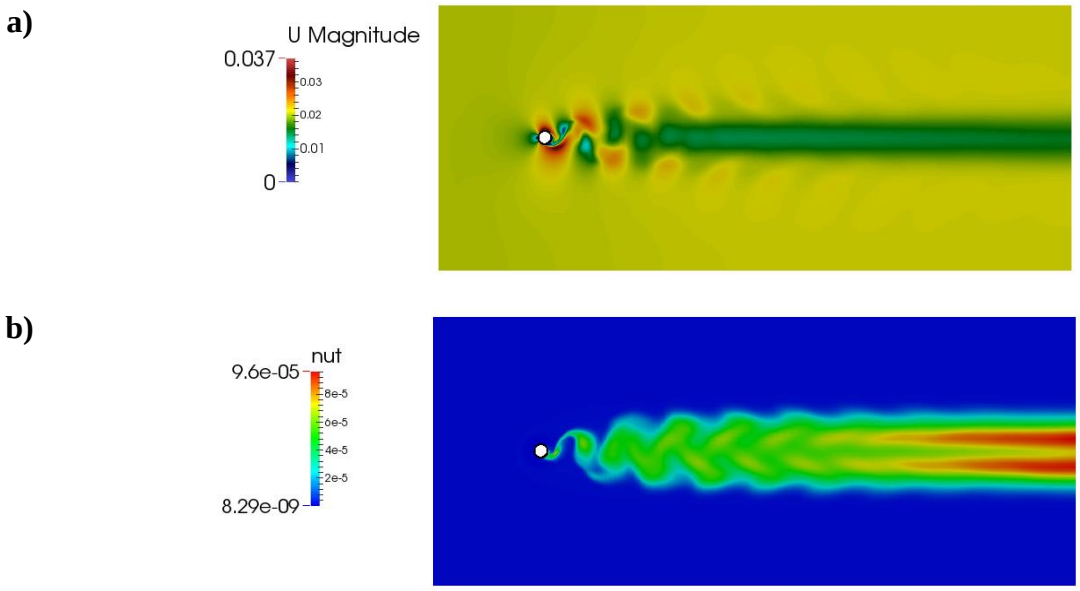


Figure 5.23: Whole domain visualization in paraView

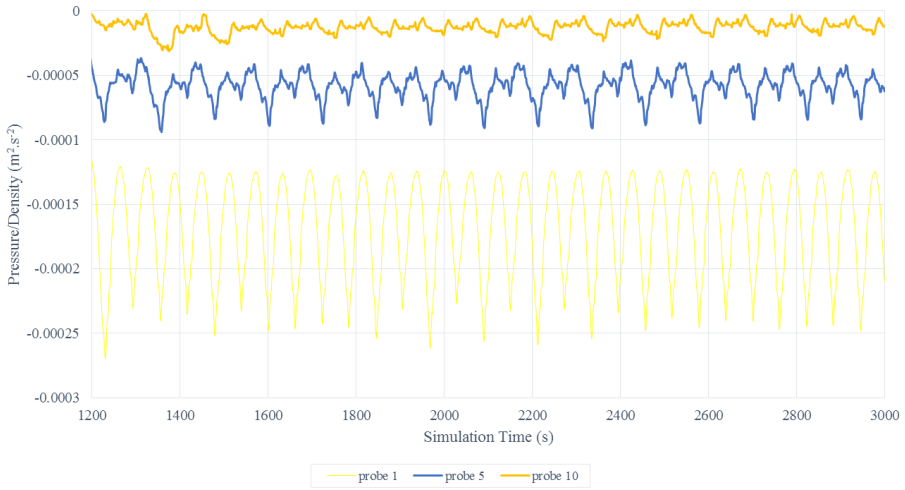


Figure 5.24: Pressure time histories of turbulent flow with Re = 10,000

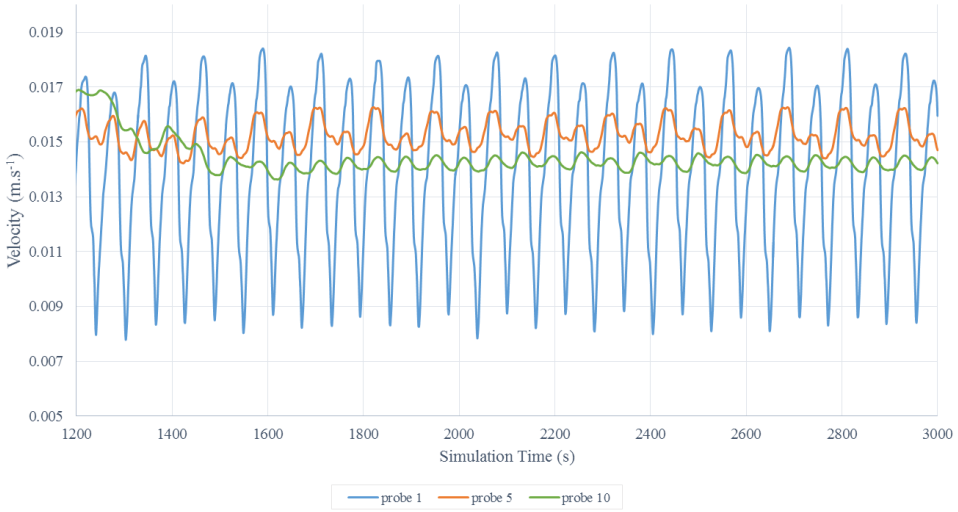


Figure 5.25: Velocity time histories of turbulent flow with Re = 10,000

The force coefficient time histories is presented in Figure 5.26. Since the oscillation of the velocity and pressure is quite random due to turbulence behavior, it is difficult measure the vortex shedding frequency from it. Therefore, in here we will calculate the vortex shedding frequency by observing the oscillation of the lift coefficient, which is presented in Figure 5.26.

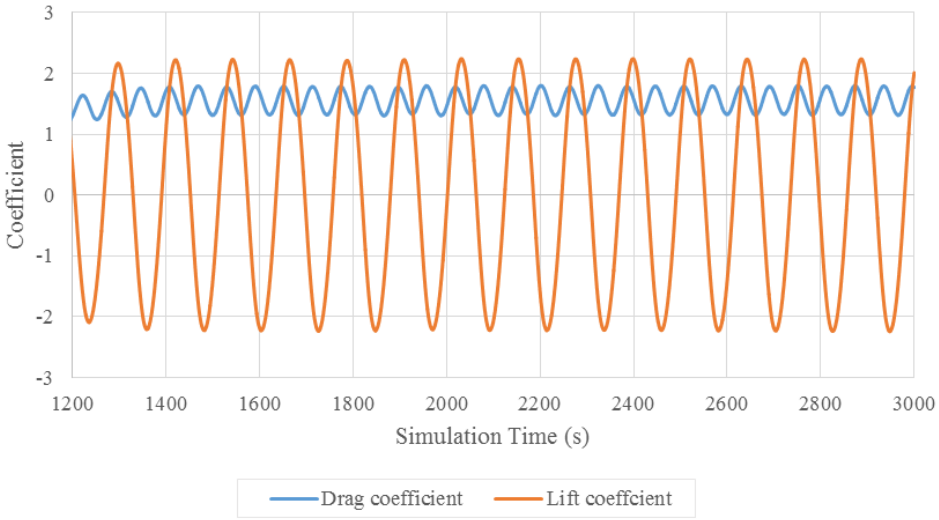


Figure 5.26: Force coefficient time histories of turbulent flow with Re = 10,000

In the end, we get vortex shedding frequency equal to 0.008 Hz and Strouhal number equal to 0.205. The mean drag coefficient is 1.55 and the amplitude of the lift coefficient is 2.24.

5.2.2. Summary and Discussion of the Turbulent Flow Simulation

The results for the remaining cases in turbulent flow simulation is presented in Appendix G. The summary for the hydrodynamic coefficients, vortex shedding frequency, and the Strouhal number of the turbulent flow case is presented in Table 5.4. Conclusively, the obtained relationship between the Strouhal and Reynolds number is similar to Figure 2.5, where the Strouhal number for rough surface (no-slip condition on the pipe's surface) is around 0.2 for Re equal to 10^4 and 10^5. However, the Strouhal number decreased slightly on $Re = 10^6$ where, according to Figure 2.5, it is supposed to be increasing instead. However, it goes up again in $Re = 10^7$.

Table 5.4: Summary of turbulent flow simulation results

Re	C_D	C_L	f_v	St
10,000	1.546 ± 0.258	2.243	0.008	0.205
100,000	1.753 ± 0.378	2.513	0.082	0.206
300,000	1.505 ± 0.359	2.469	0.242	0.201
700,000	1.551 ± 0.365	2.495	0.568	0.202
1,000,000	1.672 ± 0.389	2.455	0.788	0.196
10,000,000	1.725 ± 0.385	2.501	8.118	0.202

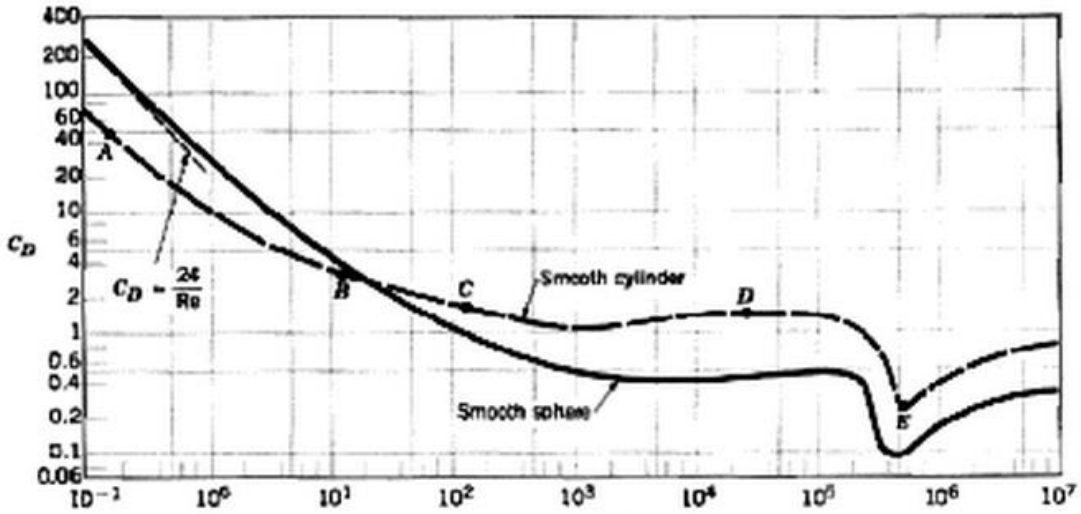


Figure 5.27: Drag coefficient as a function of Reynolds number for smooth circular cylinder and smooth spheres ("Drag of blunt bodies and streamlined bodies," n.d.)

Another contradiction occurs on the hydrodynamic coefficients, where the value of the drag coefficient of the simulation is much higher than the experiments in Figure 5.27. However, the drag coefficients from the simulation are fluctuating in the same manner with Figure 5.27 where

there is a sudden drop in the drag coefficient somewhere in the range of $10^5 < Re < 10^6$. According to Cengel and Cimbala (2010), it usually happens at about 2 x 10^5. As seen in the Table 5.4, the drag coefficient starts to drop significantly at 3 x 10^5.

Cengel and Cimbala (2010) explained that the large reduction in drag coefficient is caused by the flow in the boundary that is becoming turbulent, which moves the separation point further onto the rear of the body, reducing the size of the wake and thus the magnitude of the pressure drag. This event is in contrast to streamlined bodies, where the drag coefficient increases when the boundary layer becomes turbulent.

The difference in the drag coefficients might be caused by an incorrect simulation procedure. One thing that might cause is the fact that the turbulent simulation in this thesis has not reached the steady state. According to Lin, Zhao, and Song (2011), the computation should be integrated until it reaches steady state. Lin et al. (2011) did the turbulent simulation for Re = 10^6 with different numerical schemes in OpenFOAM. In the end, the drag coefficient for his simulation is in the range of 0.25 to 0.45, which is in accordance to Figure 5.27.

Conclusively, the turbulent flow simulation still needs more improvement until it reaches a good agreement with the other experimental results. This could be solved by applying a different turbulence model such as LES since turbulent flow is dealing with sophisticated vertices.

5.3. Pipe With Seabed Proximity Simulation Results and Discussions

5.3.1. Results and Discussion for The Case with Gap Ratio = 1

The visualization of the horizontal velocity and pressure profile are presented in Figure 5.28 and Figure 5.29 respectively.

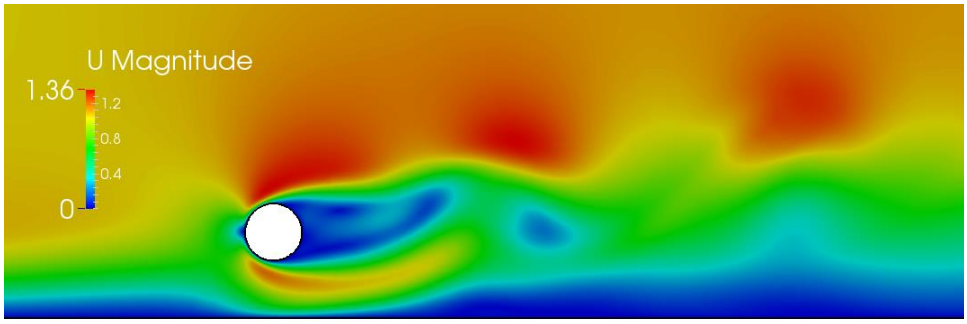


Figure 5.28: Velocity profile for case e/D = 1 in paraView

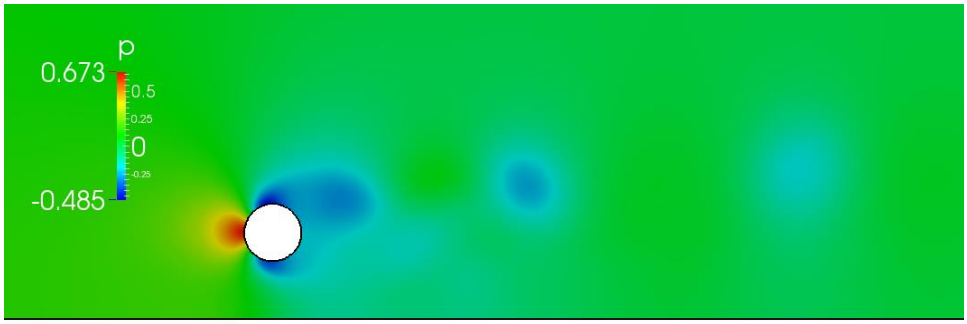


Figure 5.29: Pressure profile for case e/D = 1 in paraView

The pressure and velocity time histories are presented in Figure 5.30 and Figure 5.31. It can be seen that at this gap ratio the vortex is still oscillating. Therefore, for this case, we can still measure the vortex shedding frequency and calculate the Strouhal number.

The force coefficient time histories are presented in Figure 5.32. It can be seen from the figure that the mean lift coefficient has negative value, which means that the pipe is being drown into the seabed. If we take a look at Figure 5.28, the velocity below the pipe becomes faster than the velocity above the pipe due to the gap and the no-slip condition. Therefore, there will be more pressure that is pushing down the pipe due to slower velocity on the top side of the pipe and it will result in a negative resultant force. Consequently, the mean lift coefficient will be negative as well.

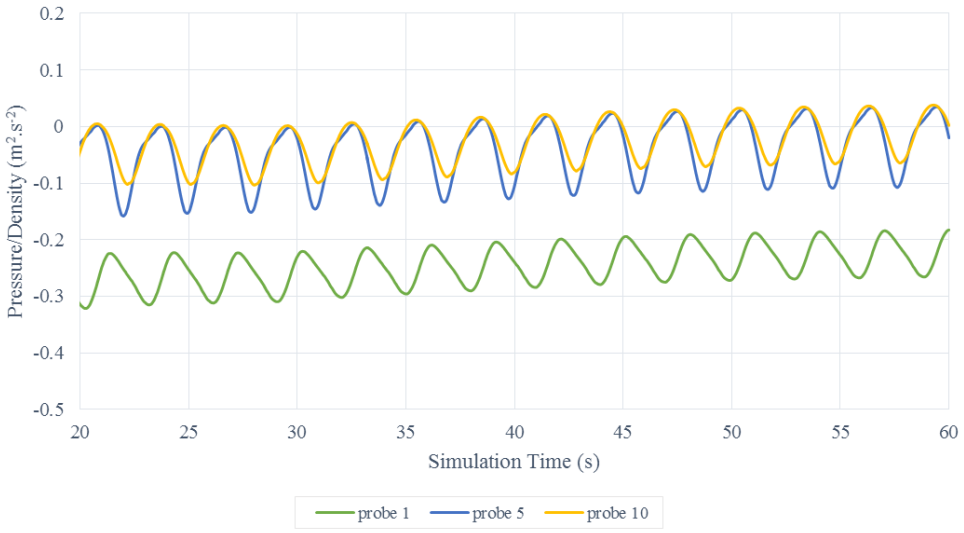


Figure 5.30: Pressure time histories for case e/D = 1

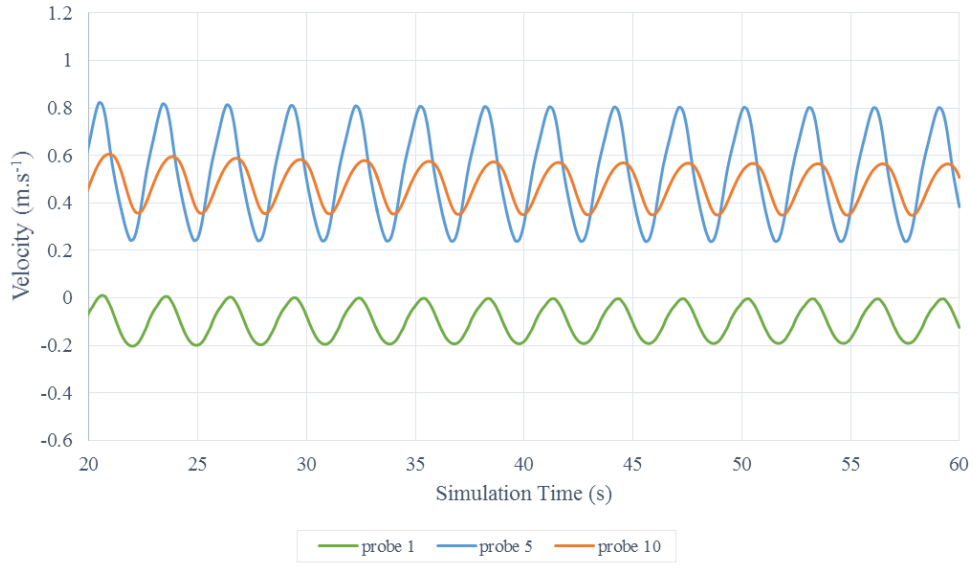


Figure 5.31: Velocity time histories for case e/D = 1

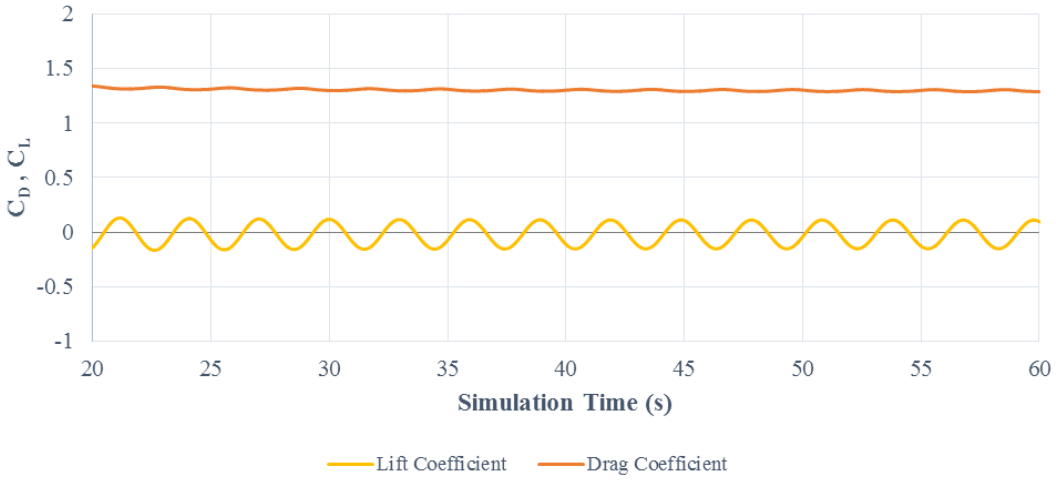


Figure 5.32: Force coefficient time histories for case e/D = 1

5.3.2. Results and Discussions for Pipe on Seabed Case (Gap Ratio = 0)

The visualization of the horizontal velocity and pressure profile are presented in Figure 5.33 and Figure 5.34 respectively.

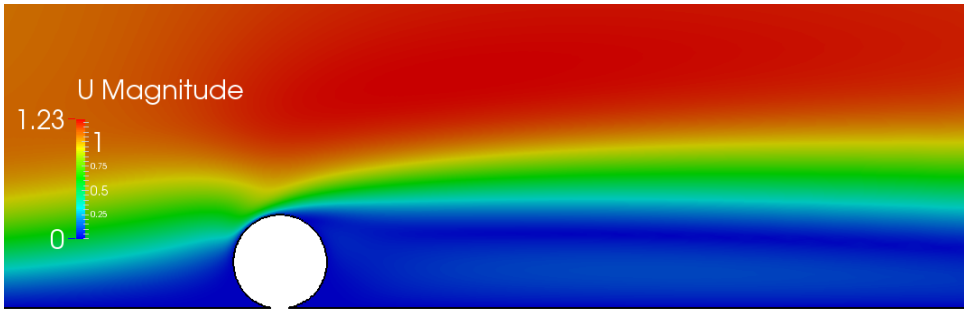


Figure 5.33: Velocity profile for case e/D = 0 in paraView

Figure 5.34: Pressure profile for case e/D = 0 in paraView

The velocity and pressure time histories are presented in Figure 5.35 and Figure 5.36 respectively. It can be seen that at this gap ratio there is no occurrence of vortex shedding

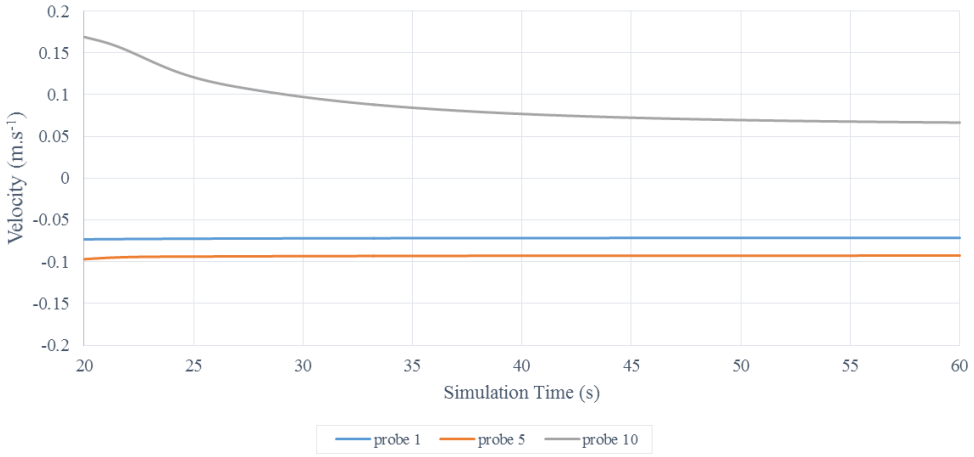


Figure 5.35: Velocity time histories for case e/D = 0

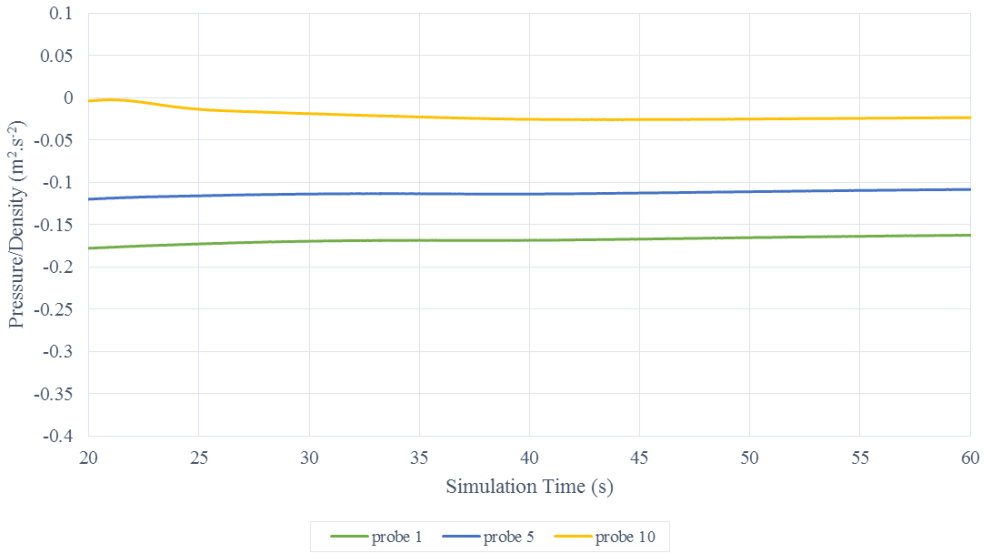


Figure 5.36: Pressure time histories for case e/D = 0

The hydrodynamic coefficient time histories are presented in Figure 5.37. It can be seen that there is no oscillation for both drag and lift coefficient. Furthermore, the lift coefficient now has positive non-zero mean value and has increased significantly compared with the case where the gap ratio is equal to 1.

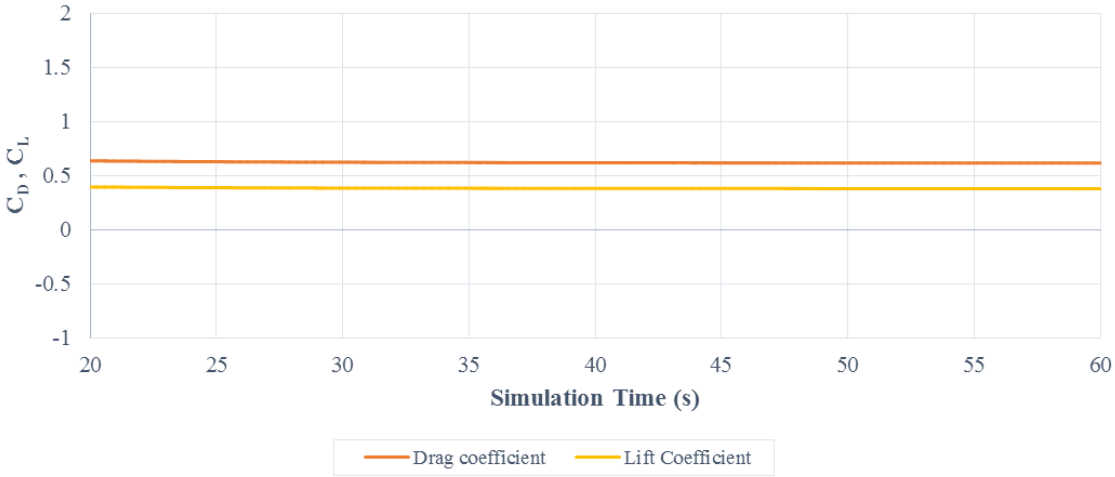


Figure 5.37: Hydrodynamic coefficient time histories for case e/D = 0

5.3.3. Summary and Discussion for Pipe with Seabed Proximity Case

The remaining simulation results of a pipe with seabed proximity are presented in Appendix G. Parameters such as hydrodynamic coefficients, Strouhal number, and vortex shedding frequency

are summarized in Table 5.5. Based on the summary, the vortex shedding ceased to exist after the gap ratio reaches 0.5 and below.

Table 5.5: Summary of pipe with seabed proximity simulations

Gap ratio	*St*	f_v	C_L			C_D		
			min	average	max			
Free Pipe	0.163	0.326	-1.009	0.000	1.009	1.410	±	0.101
1	0.184	0.369	-0.153	-0.013	0.111	1.295	±	0.011
0.5	-		0.052			0.933		
0.25			0.327			0.745		
0			0.379			0.616		

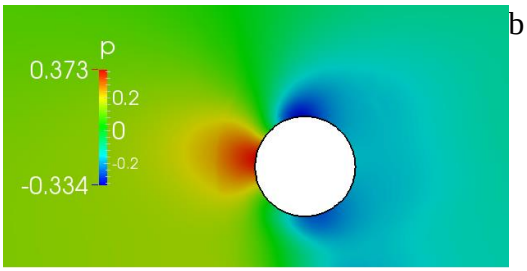
The most significant changes happen to the hydrodynamic coefficients, where the lift coefficient is now having a non-zero mean value and the drag coefficient keeps decreasing as the gap ratio decreases. In pipe with seabed proximity case, shear flow will exist because of the no-slip condition that was applied on the seabed. And as the gap ratio decreases, the wake will be formed in both front side and back side of the pipe due to the shear flow. As the gap ratio decreases, the wake in the front side of the pipe will become larger and thus result in a smaller horizontal resultant force. In another words, the drag coefficient will decrease as the gap ratio decreases

On the other hand, Sumer and Fredsøe (1997) stated,

> *"When the cylinder is moved extremely close to the wall, more and more fluid will be diverted to pass over the cylinder, which will lead to larger and larger suction pressure on the free-stream side of the cylinder. When the cylinder is sitting on the wall, the suction pressure on the cylinder surface will be the largest"*

Additionally, it was also stated that the stagnation point will move to lower and lower angular positions as the gap ratio is decreased. The simulation results agree with this remark as seen in Figure 5.38.

a)



b)

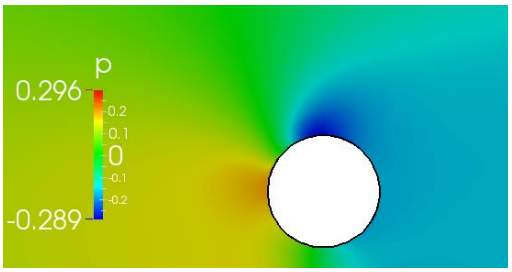


Figure 5.38: Stagnation point of the simulation results at a) e/D=0.5 b) e/D=0.25

Therefore, the combination of the stagnation point's movement and larger suction on the free-stream side of the cylinder will result in larger and larger lift forces as the pipe is moving towards the wall.

However, according to Figure 5.39, the lift coefficient for higher Reynolds number should have been decreased at a certain small gap ratio (probably around e/D = 0.05). Even though the Reynolds number that was simulated in this thesis is at a different value with Figure 5.39, the relationship between the lift and Reynolds number should've been similar to Figure 5.39. Unfortunately, this relationship could not be captured in the simulation because only a few cases were done in this thesis (lack of simulation data).

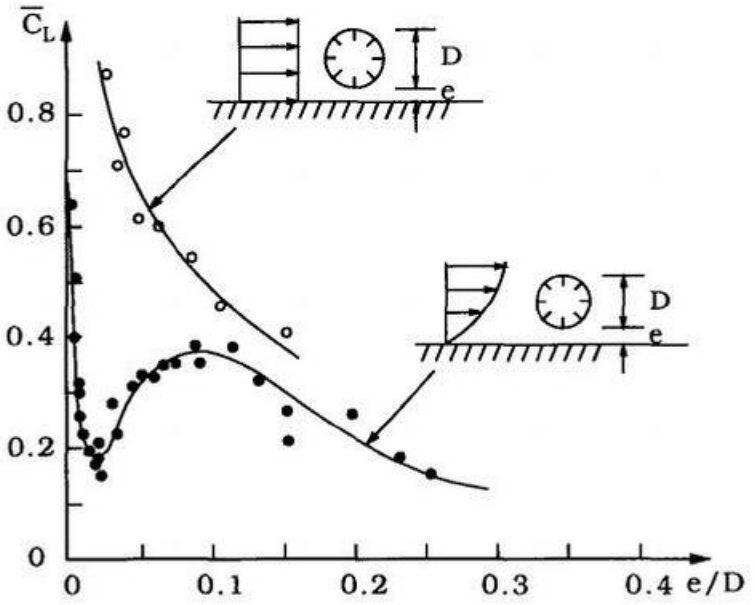


Figure 5.39: Comparison of lift coefficient in shear-free and shear flows, $10^3 < Re < 3x10^4$ (Sumer & Fredsøe, 1997)

CHAPTER 6 - Vortex Induced Vibration of the Pipe

The vortex induced vibration (VIV) simulation of the pipe was done for laminar flow with Reynolds number equal to 500. As stated by Gavin (2014), the representation for one degree of freedom system is a spring-mass-damper system (as illustrated in Figure 2.10) where the spring has no damping or mass, the mass has no stiffness or damping, the damper has no stiffness or mass. However, for this thesis, there will be no structural damping included in the motion of the pipe as it is assumed that the damping is provided by the fluid due to the viscosity.

6.1. Case Setup

6.1.1. Mesh Generation

Since the pipe will be moving due to the vortex shedding, we need to define a static mesh and a dynamic mesh that is able to move along with the pipe's motion. For this purpose, we need an additional file to define the motion of the mesh, which is the *dynamicMeshDict.* This file is placed under the constant case directory, as described in Figure 6.1.

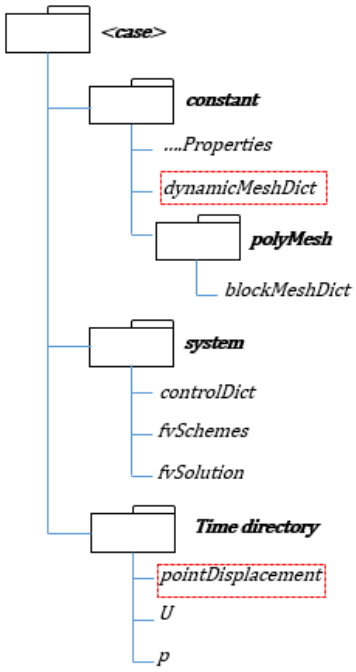


Figure 6.1: Case files for VIV case

Firstly, the static mesh is defined by *blockMeshDict* and it has the same topology with Figure 4.2. The size of the domain in this simulation is L_u = 22D, L_d = 30D, H = 22D. The mesh used in the VIV case is described in Figure 6.2. As seen in the figure, the mesh near the pipe's wall is made to be fine enough so that it can capture the vortex shedding, which is necessary to trigger the pipe's motion. Other than the near wall region, the mesh is made to be coarse to minimize the running time.

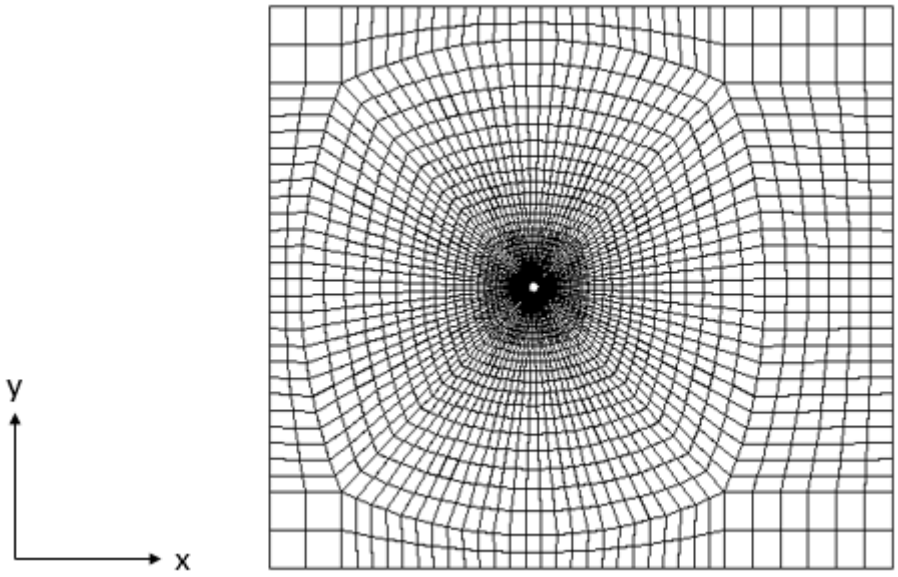


Figure 6.2: The mesh of the domain in *VIV* case

Figure 6.3 describes the content of the *dynamicMeshDict* and Table 6.1 explains its content.

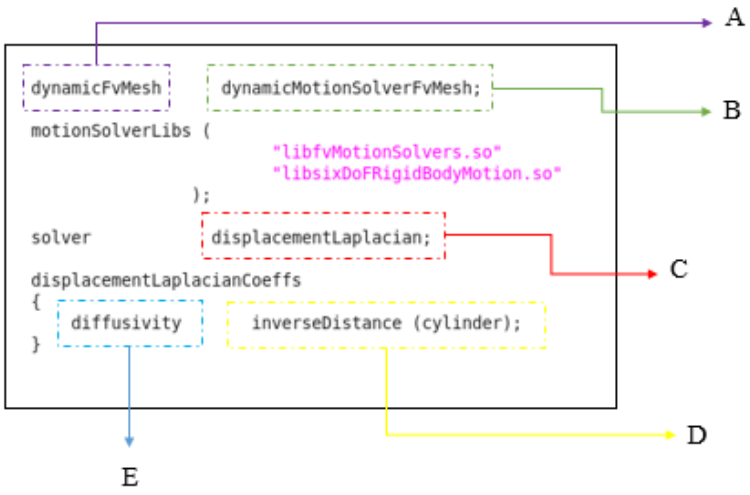


Figure 6.3: The content of *dynamicMeshDict* file

Table 6.1: Explanation about the content of *dynamicMeshDict* (González, 2009)

Symbol	Definition
A	Automatic mesh motion for a case where the topology of the mesh does not change.
B	The simplest type of mesh motion solver. For cases where the motion of interval points is solved by using boundary condition and diffusivity models.
C	The equations of cell motion are solved based on the Laplacian on the diffusivity and the cell displacement. The cell displacement is defined by *pointDisplacement* in the time directories folder.
D	One or more boundaries are specified and the diffusivity of the field is based on the inverse of the distance from that boundary.
E	Determines how the points should be moved when solving the cell equation for each time step.

6.1.2. Boundary condition

The initial and/or boundary condition should be defined for the velocity, pressure, and the position of the pipe. The initial conditions for the velocity and pressure have been explained previously in the section 4.1.4.1.3.4.1.3.1. The initial condition of the pipe's position is defined in the `pointDisplacement` file (Appendix B.6). There are many things that should be defined in the file such as follows:

- The freedom of the movement of the rigid body (cylinder). In this case, the *sixDoFRigidBodyDisplacemen*t is adopted, where the movement of the rigid body could be set in six degrees of freedom movements
- The mass of the rigid body (pipe)
- The position of the center of mass of the pipe, which in this case is defined to be located in the coordinate x = 0, y = 0, and z = 0
- The initial position of the pipe, which in this case is located in the same coordinate with the center of mass
- The moment of inertia of the rigid body, which in this case has the shape of a thin-walled pipe. The moment of inertia measures the extent to which an object resists rotational acceleration about a particular axis. The moment of inertia for thin-walled pipe, according to schematic illustration in Figure 6.4, is formulated as follows

$$I_z = \frac{1}{2} m \left(r_1^2 + r_2^2 \right) \tag{67}$$

$$I_x = I_y = \frac{1}{12} m\left[3\left(r_1^2 + r_2^2\right) + h^2\right] \quad (68)$$

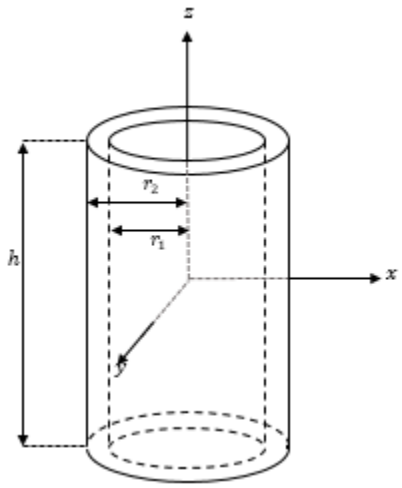


Figure 6.4: Thin-walled cylindrical tube

- The gravitational acceleration and the density of the fluid, which are 9.81 ms^{-2} and 1024 $kg.m^{-3}$ respectively.
- The velocity, acceleration, angular momentum, and torque that is applied on the rigid body. In this thesis, these parameters are defined as zero since the only external force that will be applied to the cylinder comes from the fluid flow (hydrodynamics)
- Constraints, which define the degree of freedom of the rigid body. Generally, a real structure has a motion with six degrees of freedoms which include 3 translational and 3 rotational movements, as seen in Figure 6.5. In this thesis, the pipe is defined to have only one degree of freedom, which is a vertical translation (heave).

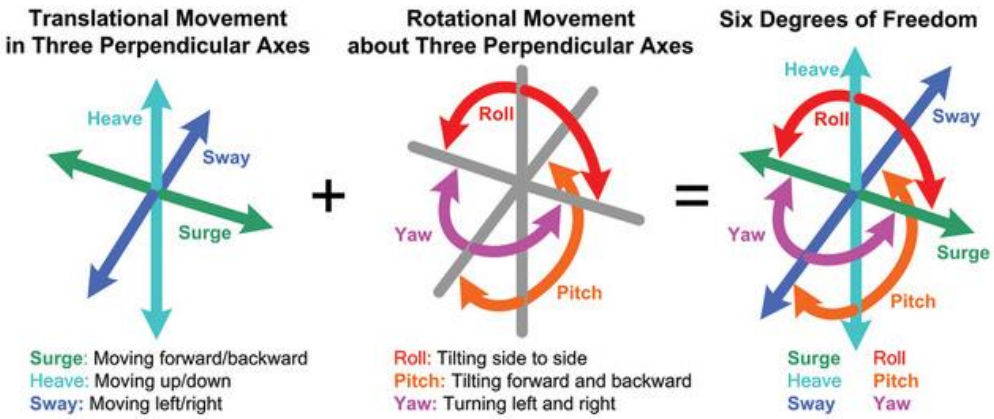


Figure 6.5: Six degrees of freedom of a system (Nelson, 2013)

- Restraints, which represents the structural property of the pipe. In this thesis, the stiffness of the pipe is represented by a vertical spring below the pipe, as seen in Figure 6.6. For the vertical spring, we shall define the damping, stiffness, rest length, restraint attachment point, and anchor's location. In this case, the damping of the spring is defined as zero since it is assumed that the damping comes from the viscous fluid, not the structure itself.

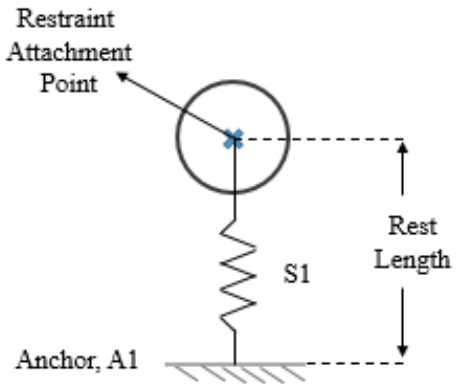


Figure 6.6: Illustration of the vertical spring below the pipe that acts as a restraint.

6.1.3. Solution Setup

6.1.3.1. Time and Data Input/Output

The time step should always be adjusted so that the Courant number is always below 0.2 and gives accurate results. For the VIV cases, the time step is set as 0.001 s and the simulation time is always set to be 90 s. This setting gives low Courant numbers and thus it can be concluded that the results are accurate. The `controlDict` directory of VIV simulation is presented in Appendix C.3.

6.1.3.2. Discretization scheme

The `fvSchemes` dictionaries of VIV simulation are presented in Appendix D.3 The sub-dictionaries of `fvSchemes` in the VIV simulation are as follows:

- `fvSchemes/ddtSchemes`: the time derivative scheme is set as `Euler`, which is a bounded, implicit scheme that has 1st order accuracy.
- `fvSchemes/divSchemes`: in this case, the discretization scheme is using Gauss scheme and the interpolation scheme is linearUpWind for the velocity, and the rest are set as `linear` for the flux and `limitedLinear` for the convection terms.

- `fvSchemes/gradSchemes`: the discretization method is set as *Gauss*. The interpolation scheme is employed as `linear`.
- `fvSchemes/snGradSchemes`: the scheme is set as `uncorrected` which means that the non-orthogonal correction is not employed.
- `fvSchemes/laplacianSchemes`: the discretization method for all variables is set as `Gauss`, the interpolation scheme is `linear`, and the surface normal gradient scheme is set as `limited` with a grade of correction of 0.5.
- `fvSchemes/interpolationSchemes`: the interpolation method for the velocity is set as `linear` which has 2nd order accuracy and unbounded
- `fvSchemes/fluxRequired`: the flux is generated from pressure since the fluxes are computed after solving the pressure equation (B. H. Hjertager, 2009).

6.1.3.3. Solution and Algorithm Control

The smoothing scheme that is applied in the solver is `GaussSeidel`. The `GAMG` is applied to pressure and point displacement, while `PBiCG` is applied to the velocity to avoid convergence problems. The pressure-velocity coupling method in VIV case is `PIMPLE`, which is a combination of `PISO` and `SIMPLE`. The `fvSolution` directory of VIV simulation is presented in Appendix E.3.

6.1.3.4. Solver

The VIV simulation will be conducted by using `pimpleDymFoam`. This solver has the same function as `pimpleFoam`, the only difference is that `pimpleDymFoam` is used for mesh that is dynamically moving. The `pimpleFoam` itself, is a large time-step solver incompressible flow using the PIMPLE (merged PISO-SIMPLE) algorithm.

6.1.4. Troubleshooting the Resonance Case

Doing the VIV simulation was not always a smooth process, especially for the case where the pipe is in resonance. In this case, the vertical displacement of the pipe was very large and the pipe was oddly in a roll motion even though the pipe was originally enabled to move in vertical translation only. Since we want to observe the vortex shedding and oscillation frequencies of this case as well, several actions need to be done in order to capture the pipe's motion during resonance. The encountered problems and its solutions are elaborated in Appendix F.

6.2. Results of VIV Simulation

The results of the VIV simulation are divided into 2 parts, which are the non-resonance case and resonance case.

6.2.1. Non-resonance case

A case where the mass of the pipe has the value of 100 kg and its stiffness is equal to 1750 N/m had been simulated. In this case, the resulted value of Eigen frequency is 0.608 Hz. Figure 6.7 shows the comparison of the non-dimensional vertical displacement (*y/D*) and lift coefficient time histories of the pipe. It can be seen that the pipe's response arises along with the vortex oscillation, which indicates a Vortex Induced Vibration (VIV) of the pipe.

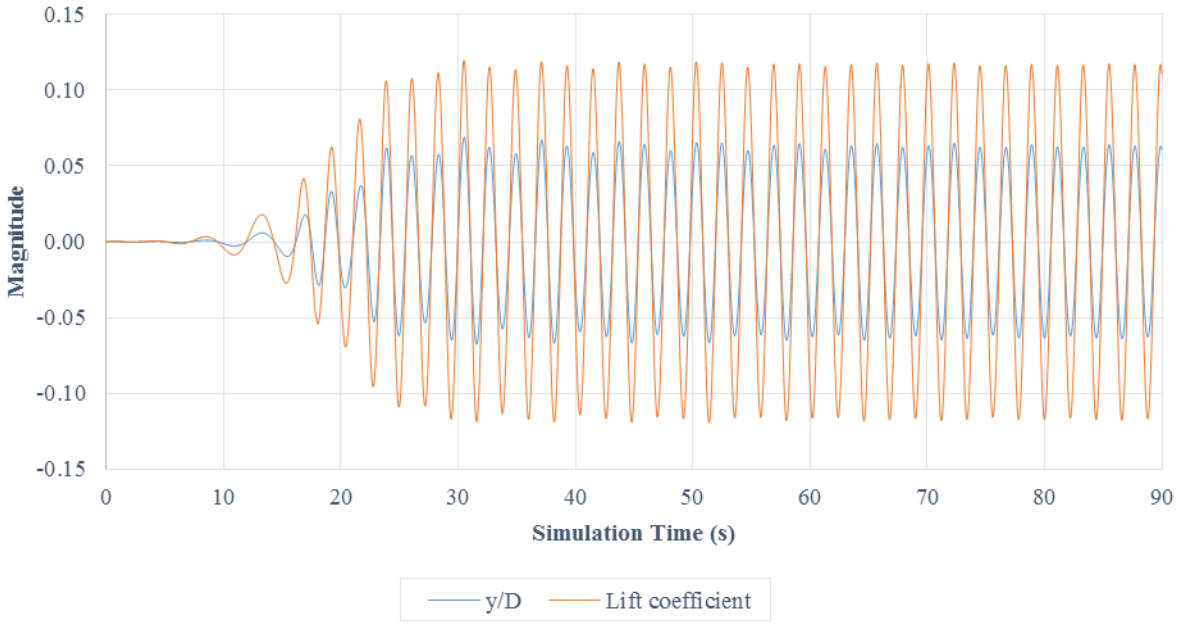
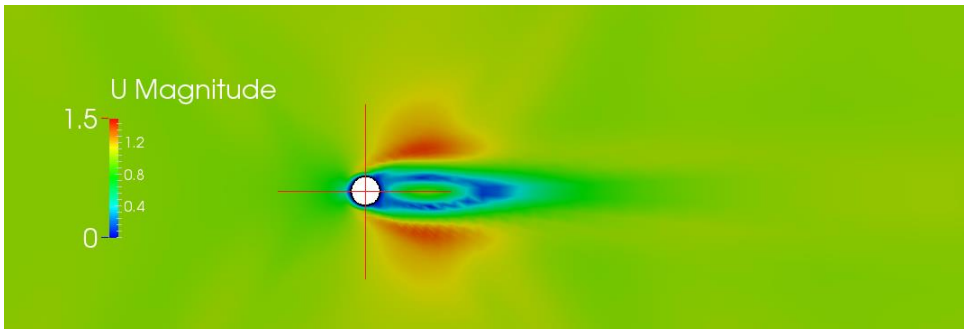
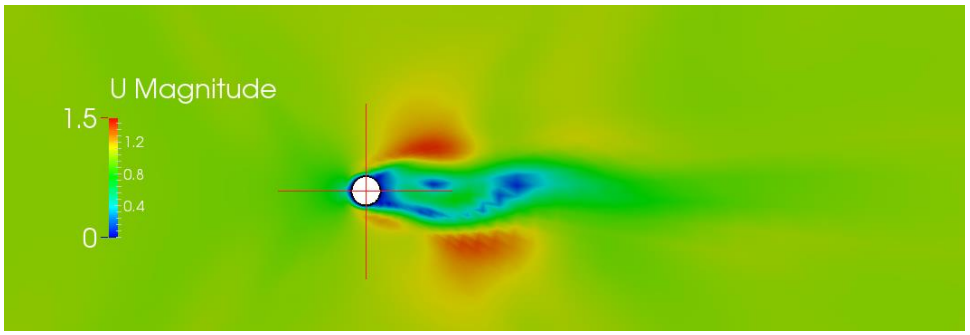
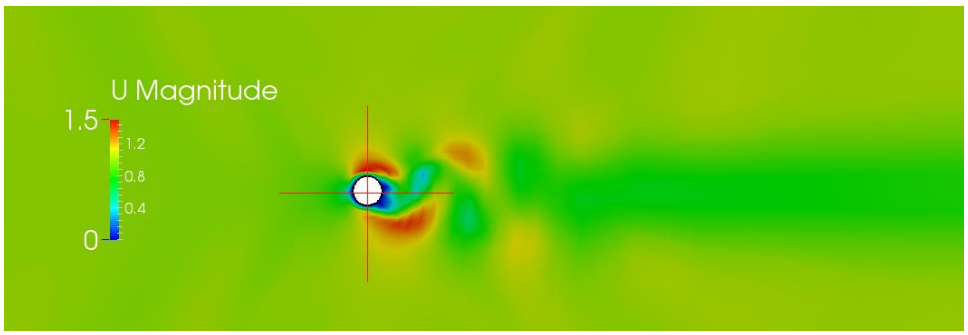


Figure 6.7: Lift coefficient and displacement time histories of the pipe at Re = 500

The frequency of the vortex shedding behind the pipe, f_v, has the value of 0.909 Hz, while the oscillation frequency, f_{osc}, has the value of 0.454 Hz. Eventually, the flow regime behind the pipe is similar to the turbulent flow regime, where the oscillating wake behind the pipe becomes smaller in size. The flow regime of this case can be seen in Table 6.2.

Table 6.2: Flow regime around the oscillating pipe (non-resonance case)

Flow Regime Around the Oscillating Pipe
 t = 10s
 t = 15 s
 t = 90 s

6.2.2. Resonance case

Another simulation was done for the same Reynolds number. In this case, the mass of the pipe has the value of 180 kg and the stiffness of 1750 N/m. Therefore, the resulted value of Eigen frequency is 0.471 Hz. Initially, this case ended up in a very high vertical displacement and the simulation stopped at the beginning due to the overly displaced mesh. This probably indicated that the pipe is in resonance. Afterwards, several actions were done (refer to section 6.1.4) so that the simulation can still go on and parameters such as vortex shedding and oscillation frequencies can still be

captured. Figure 6.8 shows the comparison of the non-dimensional vertical displacement and lift coefficient time histories of the pipe.

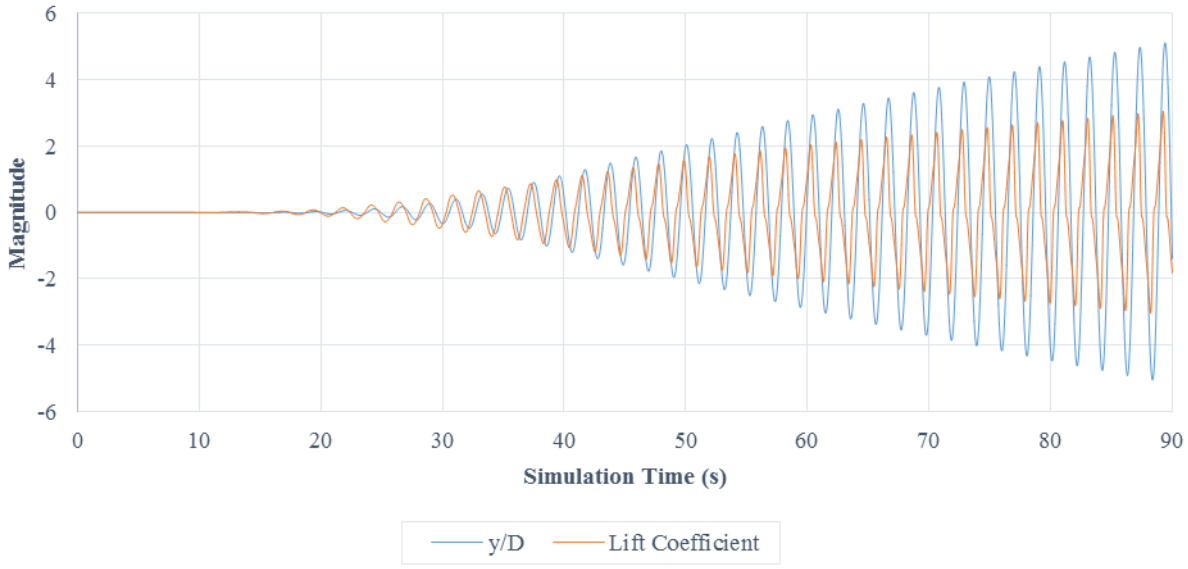


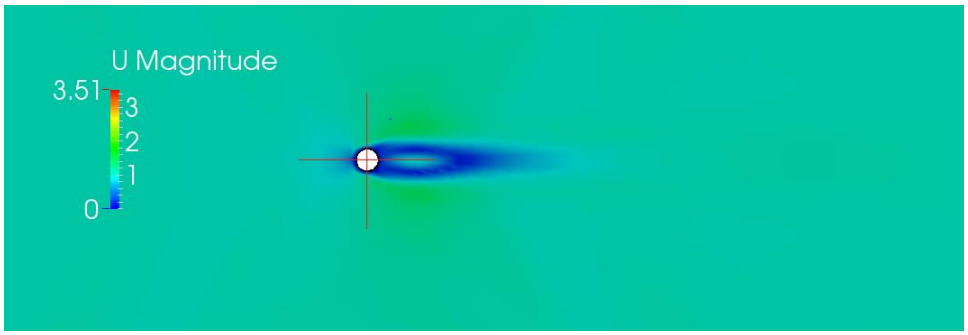
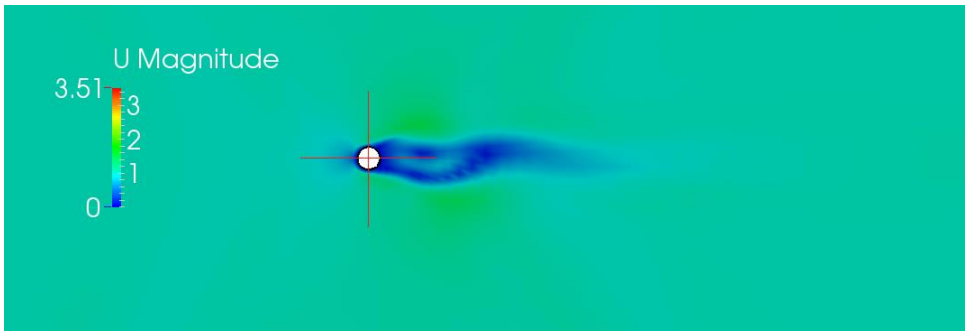
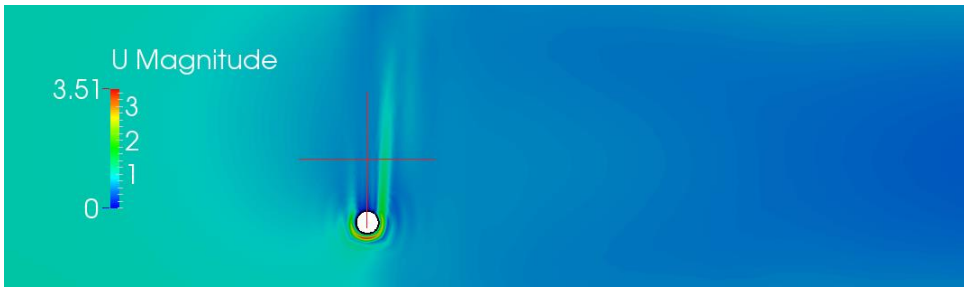
Figure 6.8: Lift coefficient and displacement time histories of the resonance case

The frequency of the vortex shedding behind the pipe, f_v, has the value of 0.488 Hz, while the oscillation frequency, f_{osc}, has the value of 0.485 Hz.

It can be seen that the magnitude of the non-dimensional vertical displacement is very big and keeps escalating as the simulation time goes by. At the end, the magnitude of the non-dimensional vertical displacement is higher than the lift coefficient. These results are in contrast with the results of the non-resonance case in Figure 6.7.

Additionally, the flow regime of this case does not represent either the laminar or turbulent flow since the wake behind the pipe cannot be maintained due to the high amplitude oscillation. The flow regime of this case can be seen in Table 6.3.

Table 6.3: Flow regime around the oscillating pipe (resonance case)

Flow Regime Around the Oscillating Pipe
 t = 10 s
 t = 15 s
 t = 90 s

6.3. Results Summary and Discussion

There is a total of 5 VIV cases with a different Eigen frequency that were simulated in this thesis. The rest of the simulation results are presented in Appendix G. The Eigen frequency was modified by either changing the mass of the pipe or its stiffness. Table 6.4 shows the results summary of all cases in the VIV simulation. The parameters that are presented for each case in the Table 6.4 are mass (m), stiffness (k), Eigen frequency (f_n), oscillation frequency (f_{osc}), vortex shedding frequency (f_v), frequency ratio (β), oscillation amplitude (A), and reduced velocity (V_R). The reduced velocity is calculated based on the Eigen frequency in the still water, f_n.

Table 6.4: Results summary of the VIV simulation

Case	m	k	f_n	f_{osc}	f_v	β	A	V_R
1	100	1750	0.608	0.454	0.909	0.748	0.034	3.292
2	100	2000	0.649	0.430	0.881	0.661	0.022	3.079
3	90	1750	0.635	0.441	0.898	0.695	0.029	3.152
4	110	1750	0.584	0.468	0.934	0.801	0.043	3.426
5	180	1750	0.471	0.485	0.488	1.030	5.083	4.249

Case number 1 to 4 are non-resonance cases, while case number 5 is a resonance case. We shall note that the oscillation amplitude in the resonance case is not its actual amplitude since damping was applied in the vertical spring of this case. However, we can see that even though the amplitude has been reduced already by the damper, it still has much larger value than the other case.

Figure 6.9 shows the relationship of frequencies such as the oscillation, vortex shedding, and Eigen frequency with the reduced velocity.

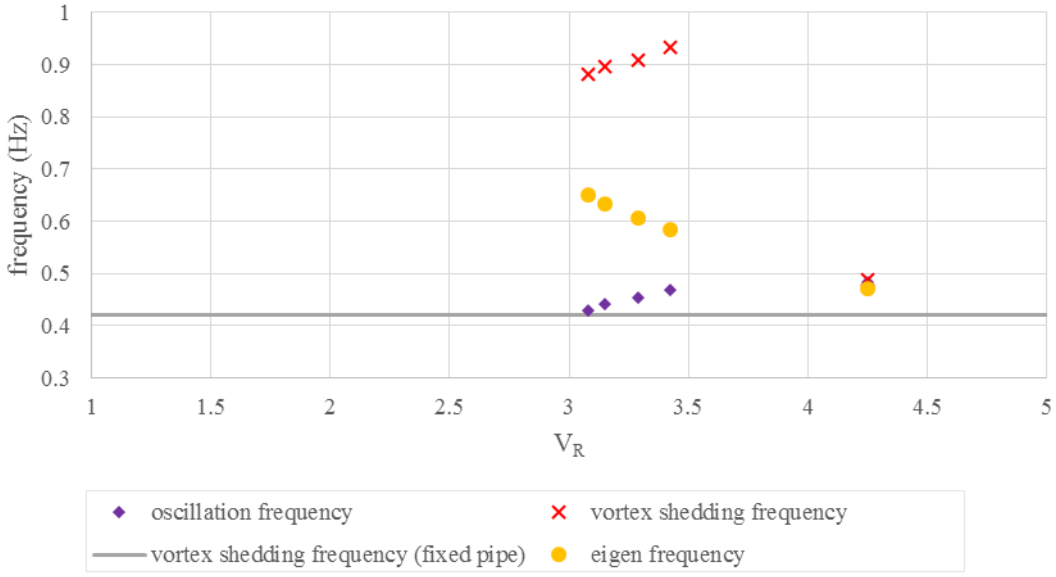


Figure 6.9: Frequency – reduced velocity relationship

It can be seen that when the reduced velocity has the value of 4.249 (case number 5), the oscillation frequency, vortex shedding frequency, and Eigen frequency fall into the same value ("lock-in" phenomena). As explained previously in section 2.6.3 (page 17), the "lock-in" phenomena will result in a very high Dynamic Amplification Factor (DAF). This is the reason why the amplitude

of case number 5 is tremendously higher than the other cases. Conclusively, the case number 5 is a resonance case. Additionally, it can be seen that the oscillation frequency always stays near the vortex shedding frequency of the fixed pipe. Therefore, it can be concluded that in order to avoid resonance we shall not have a natural frequency that has an approximate value to the vortex shedding frequency of the fixed pipe.

According to the experiment that was done by Feng (1968), the resonance should have happened when the reduced velocity is in the range of 5 to 7. While in the present study, the resonance had already happened when the reduced velocity reached 4.249 (refer to Figure 6.10). Apparently, similar results were also obtained by Vikestad, Vandiver, and Larsen (2000) where their results showed that the "lock-in" occurred when the reduced velocity is in between 4 and 12 (refer to Figure 6.11)

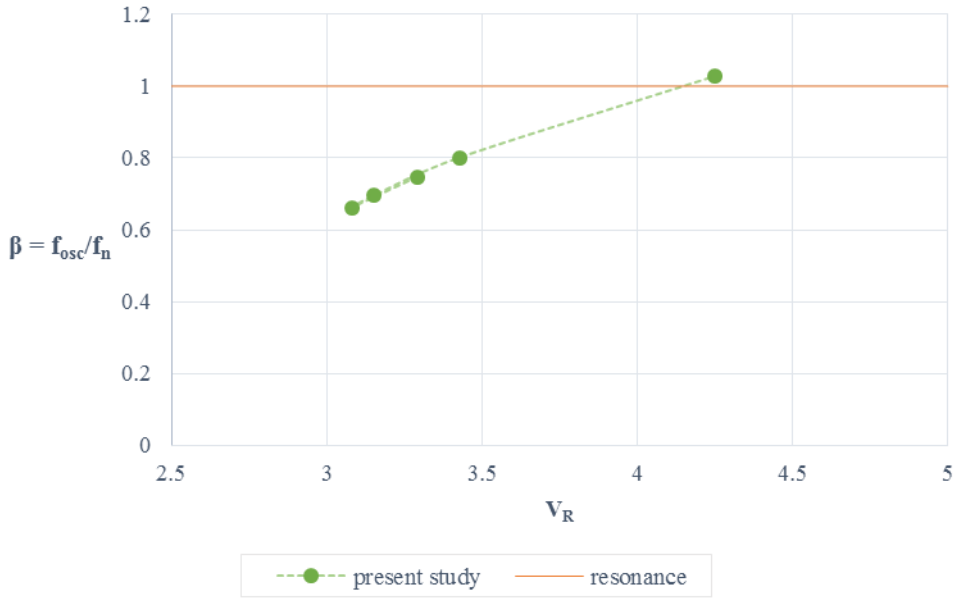


Figure 6.10: Frequency ratio – reduced velocity relationship of the present study

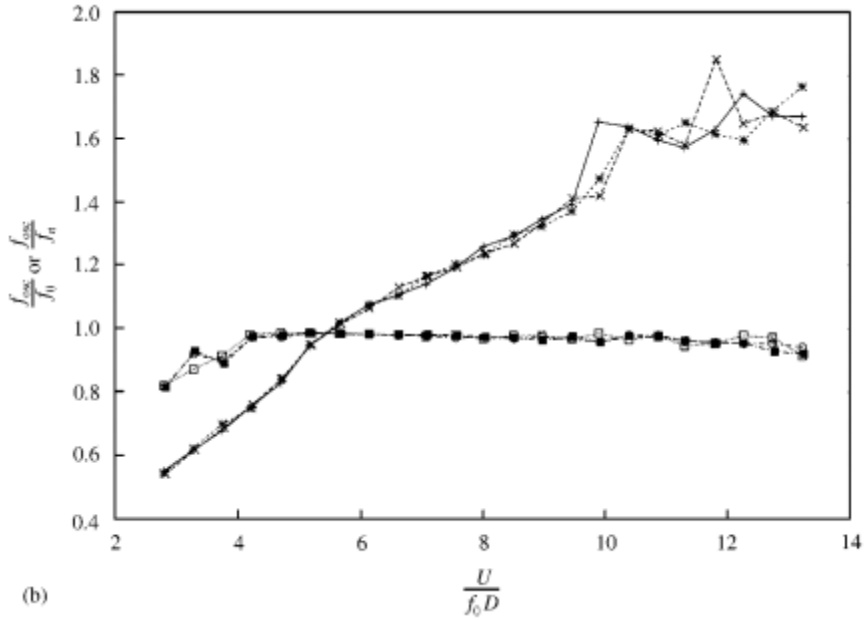


Figure 6.11: Frequency ratio – reduced velocity relationship (Vikestad et al., 2000)

Therefore, it can be concluded that the VIV simulations of this thesis have a good agreement with other experimental results.

6.4. The Effect of the Fluid Damping

The oscillation of the pipe is expected to be similar to Figure 2.16 in page 20, where a decrease in the oscillation occurs due to the fluid damping. The fluid damping will result in energy dissipation as the fluid moves relative to the oscillating pipe. The interaction between the fluid and the pipe produces added mass and supposedly affect the fluid damping. However, the simulation results show that neither Figure 6.7 nor Figure 6.8 resembles Figure 2.16 in page 20.

The same results were also obtained by Asyikin (2012) and it was suspected that the fluid damping will not take effect for VIV with small amplitude. Presumably, the small oscillation amplitude will result in small effect of the added mass and thus will reduce the effect of the fluid damping.

In the Figure 6.7, the amplitude oscillation is very small (less than 10% of the diameter) and hence the fluid damping is not dissipating the oscillation. As for the Figure 6.8, the pipe is in resonance and thus the fluid damping is probably not strong enough to dissipate the oscillation.

CHAPTER 7 - Conclusions and Suggestions

5.1. Conclusions

5.1.1. Flow Over a Fixed Pipe

5.1.1.1. Laminar Flow Simulation

- There is no vortex shedding occurrence for Re = 20, and the drag coefficient in this simulation is considerably higher due to the large wake that is created behind the pipe. There is no lift coefficient in this case because the wake is still symmetrical.
- The drag coefficients for $100 \leq Re \leq 1000$ are almost always constant as the Reynolds number increases.
- The amplitude of the lift coefficient appears to be increasing exponentially as the Reynolds number increases.
- The Strouhal – Reynolds number relationship that was gained from the simulation has a good agreement with other experimental results.

5.1.1.2. Turbulent Flow Simulation

- The wake created behind the pipe becomes thinner as the Reynolds number increases. This remark agrees with the theory explained by Cengel and Cimbala (2010).
- The Strouhal – Reynolds number relationships gained from the turbulent simulation have a good agreement with the Strouhal – Reynolds number relationships for rough cylinder in Figure 2.5, page 9.
- The drag coefficients gained from the turbulent flow are behaving in a same manner as the drag coefficients in Figure 5.27, page 71. However, the value of the drag coefficients in the simulation is much higher than expected. This fact indicates that the turbulent flow simulation is still incorrect and needs more improvement.

5.1.1.3. Pipe with Seabed Proximity Simulation

- The vortex shedding is suppressed when the gap ratio reaches 0.5.
- When the gap ratio is equal to 1 the vortex shedding still exists and it causes an increase in the vortex shedding frequency but not high enough to be concerned.
- The stagnation point moves to a lower angular position as the gap ratio is decreased.

- The lift coefficient has a non-zero mean value when the vortex shedding is suppressed at the gap ratio = 0.5
- The drag coefficient keeps decreasing as the gap ratio decreases.
- The lift coefficient – gap ratio relationship in Figure 5.39 (page 78) could not be captured by the simulation results because only a few cases were done in this thesis (lack of simulation data). Conclusively, the simulation of flow over a pipe with seabed proximity needs improvement.

5.1.2. VIV Simulation

- Modifying the mass of the cylinder or its stiffness will affect the resulting Eigen frequency. Lighter mass and stronger stiffness will result in an even higher Eigen frequency and hence we will get smaller reduced velocity.
- When the reduced velocity reaches 4, the "lock-in" phenomena will happen and thus will result in resonance.
- Conclusively, it is best to avoid the "lock-in" range, which means that we shall keep the reduced velocity to be less than 4.

5.2. Suggestions for Future Work

- Do an independent mesh quality evaluation such as trying different cell number for the simulation and then compare the results. This shall be done to find out the most effective cell number for the simulation.
- For the VIV simulation, try a wider range of reduced velocity to be tested. For example, start from a considerable small value of reduced velocity until a certain high value.
- Do the turbulent simulation using another turbulence model such as the k-epsilon or use the LES method to resolve the sophisticated turbulent within the simulation. This might solve the problems that occurred in the turbulent flow simulation of this thesis.
- Do more simulations in the range of $10^5 < Re < 10^6$ as there are supposed to be a lot of dramatic changes that happen to the drag coefficient of this Reynolds number range (refer to Figure 5.27, page 71)
- Do more simulations for gap ratio in the range of $0.2 < e/D < 0$ to capture the relationship that is given in Figure 5.39, page 78.

References

Asyikin, M. T. (2012). *CFD Simulation of Vortex Induced Vibration.* (Master Thesis), Norwegian University of Science and Technology, Trondheim, Norway. Retrieved from http://brage.bibsys.no/xmlui/handle/11250/232233

Bakker, A. (2006a). Applied computational fluid dynamics: Meshing. Retrieved May 3, 2015, from http://www.bakker.org/dartmouth06/engs150/07-mesh.pdf

Bakker, A. (2006b). Applied computational fluid dynamics: Turbulence Models. Retrieved May 3, 2015, from http://www.bakker.org/dartmouth06/engs150/10-rans.pdf

Behr, M., Hastreiter, D., Mittal, S., & Tezduyar, T. E. (1995). Incompressible flow past a circular cylinder: dependence of the computed flow field on the location of the lateral boundaries. *Computer Methods in Applied Mechanics and Engineering, 123*(1–4), 309-316. doi: http://dx.doi.org/10.1016/0045-7825(94)00736-7

Berthelsen, P. A., & Faltinsen, O. M. (2008). A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries. *Journal of Computational Physics, 227*(9), 4354-4397. doi: http://dx.doi.org/10.1016/j.jcp.2007.12.022

Cengel, Y. A., & Cimbala, J. M. (2010). *Fluid mechanics: Fundamentals and applications* (2nd ed.). Boston: McGraw-Hill.

CFD Direct. (2015). Mesh generation with blockMesh. Retrieved April 30, 2015, from http://cfd.direct/openfoam/user-guide/blockmesh/

Det Norske Veritas. (2011). Modelling and analysis of marine operations *Appendix A: Added mass coefficients* (Vol. DNV-RP-H103). Oslo: Det Norske Veritas.

Douglas, C. (2014). *PisoFoam and simpleFoam compared*. Retrieved May 21, 2015, from http://www.calumdouglas.ch/openfoam/pisofoam-simplefoam-compared/

Drag of blunt bodies and streamlined bodies. (n.d.). Retrieved June 6, 2015, from http://www.princeton.edu/~asmits/Bicycle_web/blunt.html

Feng, C. C. (1968). *The measurements of the vortex induced effects in flow past stationary and oscillating circular and D-section cylinders.* (Master Thesis), University of British Columbia, Vancouver. Retrieved from https://circle.ubc.ca/bitstream/handle/2429/35378/UBC_1969_A7%2520F45.pdf?sequence=1

Franke, R., Rodi, W., & Schönung, B. (1990). Numerical calculation of laminar vortex-shedding flow past cylinders. *Journal of Wind Engineering and Industrial Aerodynamics, 35*, 237-257.

Gavin, H. P. (2014). Vibrations of single degree of freedom systems.In *Compendium: Structural dynamics*. Durham: Duke University. Retrieved from http://people.duke.edu/~hpgavin/cee541/sdof-dyn.pdf.

Giljarhus, K. E. (2015). Turbulence and turbulence modelling *In course: Computational fluid dynamics* Stavanger: Llyod's Register Consulting

González, A. O. (2009). Procedure for defining a mesh with a motion CFD with open source software. Göteborg: Chalmers. Retrieved from http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2009/AndreuOliverGonzalez/ProjectReport_Corrected.pdf.

Greenshields, C. J. (2015). *OpenFOAM user guide*. Retrieved from http://foam.sourceforge.net/docs/Guides-a4/UserGuide.pdf

Gudmestad, O. T. (2014). Chapter 8: Dynamics of one degree of freedom systems. In *Compendium: Marine technology and operations: Theory and practice*. Stavanger: University of Stavanger

Guo, D. B., Song, D. S., Chacko, J., & Ghalambor, D. A. (2005). Chapter 1 - Introduction. In Ghalambor (Ed.), *Offshore Pipelines* (pp. 1-10). Burlington: Gulf Professional Publishing

Henderson, R. D. (1997). Nonlinear dynamics and pattern formation in turbulent wake transition. *Journal of Fluid Mechanics, 352*, 65-112. doi: 10.1017/S0022112097007465

Hjertager, B. (2009). OpenFOAM cases: How to run an openFOAM solver. In *Compendium: Computational Fluid Dynamics*. Stavanger: University of Stavanger

Hjertager, B. H. (2009). Lecture notes in openFOAM. In *Course: Computational fluid dynamics*. Stavanger: University of Stavanger

Hjertager, B. H. (2015). Lecture notes of CFD. In *Course: Computational Fluid Dynamics*. Stavanger: University of Stavanger

Karunakaran, D. (2014). Pipeline span and VIV *In Lecture notes: Pipelines and risers*. Stavanger: University of Stavanger

Koushan, K. (2009). *Vortex Induced Vibrations of Free Span Pipelines.* (Ph.D. Thesis), Norwegian University of Science and Technology, Trondheim, Norway. Retrieved from http://brage.bibsys.no/xmlui/handle/11250/237661

Lin, Z., Zhao, D., & Song, J. (2011). The influence of advection schemes and turbulence closure models on drag coefficient calculation around a circular cylinder at high Reynolds number. *Journal of Ocean University of China, 10*(3), 229-233. doi: 10.1007/s11802-011-1789-2

Mittal, S., & Raghuvanshi, A. (2001). Control of vortex shedding behind circular cylinder for flows at low Reynolds numbers. *International Journal for Numerical Methods in Fluids, 35*(4), 421-447. doi: 10.1002/1097-0363(20010228)35:4<421::AID-FLD100>3.0.CO;2-M

Nelson, B. (2013). The 6DF IMU 6-D motion variant provides six dimensional motion sensing over six degrees of freedom by sensing translation movement in three perpendicular axes (surge, heave, sway) and rotational movement in three perpendicular axes (roll, pitch, yaw) [image]. Retrieved June 4, 2015, from http://www.oemoffhighway.com/article/10979955/inertial-measurement-sensors-improve-safety-in-ag-equipment

OpenFOAMWiki. (2009). *icoFoam*. Retrieved May 2, 2015, from https://openfoamwiki.net/index.php/IcoFoam

Patnana, V. K., Bharti, R. P., & Chhabra, R. P. (2009). Two-dimensional unsteady flow of power-law fluids over a cylinder. *Chemical Engineering Science, 64*(12), 2978-2999. doi: http://dx.doi.org/10.1016/j.ces.2009.03.029

Saxena, A. (2015). *Guidelines for specification at inflow boundaries*. Retrieved May 19, 2015, from http://support.esi-cfd.com/esi-users/turb_parameters/

Sumer, B. M., & Fredsøe, J. (1997). *Hydrodynamics Around Cylindrical Structures* (Vol. 12). Singapore: World Scientific.

Sunden, B. (2011). *Vortex Shedding*. Retrieved May 3, 2015, from http://www.thermopedia.com/content/1247/

Symscape. (2009). *Reynolds-averged navier-stoke equations*. Retrieved May 3, 2015, from http://www.symscape.com/reynolds-averaged-navier-stokes-equations

The OpenFOAM Foundation. (n.d.). *Standard solvers*. Retrieved May 2, 2015, from http://www.openfoam.org/features/standard-solvers.php

Versteeg, H. K., & Malalasekera, W. (2007). *An introduction to computational fluid dynamics : the finite volume method* (2nd ed.). Harlow: Pearson/Prentice Hall.

Vikestad, K., Vandiver, J. K., & Larsen, C. M. (2000). Added mass and oscillation frequency for a circular cylinder subjected to vortex-induced vibrations and external disturbance. *Journal of Fluids and Structures, 14*(7), 1071-1088. doi: http://dx.doi.org/10.1006/jfls.2000.0308

Williamson, C. H. K., & Brown, G. L. (1998). A series in 1/√Re to represent the Strouhal–Reynolds number relationship of the cylinder wake. *Journal of Fluids and Structures, 12*(8), 1073-1085. doi: http://dx.doi.org/10.1006/jfls.1998.0184

Zhao, M., Kaja, K., Xiang, Y., & Yan, G. (2013). Vortex-induced vibration (VIV) of a circular cylinder in combined steady and oscillatory flow. *Ocean Engineering, 73*(0), 83-95. doi: http://dx.doi.org/10.1016/j.oceaneng.2013.08.006

Appendix A - Mesh Generation

A.1. blockMeshDict

```
/*--------------------------------------------------------------------------*\
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      http://www.openfoam.org               |
|    \\/     M anipulation  |                                                 |
\*--------------------------------------------------------------------------*/

FoamFile
{
    version         2.0;
    format          ascii;
    class           dictionary;
    object          blockMeshDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
convertToMeters 1;

vertices
(
    (0.1768 0.1768 -0.5) //0
    (3 3 -0.5) //1
    (-0.1768 0.1768 -0.5) //2
    (-3 3 -0.5) //3
    (-0.1768 -0.1768 -0.5) //4
    (-3 -3 -0.5) //5
    (0.1768 -0.1768 -0.5) //6
    (3 -3 -0.5) //7
    (20 -10 -0.5) //8
    (20 -3 -0.5) //9
    (20 3 -0.5) //10
    (20 10 -0.5) //11
    (3 10 -0.5) //12
    (-3 10 -0.5) //13
    (-8 10 -0.5) //14
    (-8 3 -0.5) //15
    (-8 -3 -0.5) //16
    (-8 -10 -0.5) //17
    (-3 -10 -0.5) //18
    (3 -10 -0.5) //19
    (0.1768 0.1768 0.5) //20
    (3 3 0.5) //21
    (-0.1768 0.1768 0.5) //22
    (-3 3 0.5) //23
    (-0.1768 -0.1768 0.5) //24
    (-3 -3 0.5) //25
    (0.1768 -0.1768 0.5) //26
    (3 -3 0.5) //27
    (20 -10 0.5) //28
    (20 -3 0.5) //29
    (20 3 0.5) //30
    (20 10 0.5) //31
    (3 10 0.5) //32
    (-3 10 0.5) //33
    (-8 10 0.5) //34
```

```
    (-8 3 0.5) //35
    (-8 -3 0.5) //36
    (-8 -10 0.5) //37
    (-3 -10 0.5) //38
    (3 -10 0.5) //39

);

blocks
(
    hex (6 7 1 0 26 27 21 20) (40 40 1) simpleGrading (10 1 1) //0
    hex (2 0 1 3 22 20 21 23) (40 40 1) simpleGrading (1 10 1) //1
    hex (5 4 2 3 25 24 22 23) (40 40 1) simpleGrading (0.1 1 1) //2
    hex (5 7 6 4 25 27 26 24) (40 40 1) simpleGrading (1 0.1 1) //3
    hex (7 9 10 1 27 29 30 21) (60 40 1) simpleGrading (2 1 1) //4
    hex (1 10 11 12 21 30 31 32) (60 30 1) simpleGrading (2 2 1) //5
    hex (3 1 12 13 23 21 32 33) (40 30 1) simpleGrading (1 2 1) //6
    hex (15 3 13 14 35 23 33 34) (20 30 1) simpleGrading (1 2 1) //7
    hex (16 5 3 15 36 25 23 35) (20 40 1) simpleGrading (1 1 1) //8
    hex (17 18 5 16 37 38 25 36) (20 30 1) simpleGrading (1 0.5 1) //9
    hex (18 19 7 5 38 39 27 25) (40 30 1) simpleGrading (1 0.5 1) //10
    hex (19 8 9 7 39 28 29 27) (60 30 1) simpleGrading (2 0.5 1) //11
);

edges
(
   arc 6 0 (0.25 0 -0.5) //0
   arc 0 2 (0 0.25 -0.5) //1
   arc 2 4 (-0.25 0 -0.5) //2
   arc 4 6 (0 -0.25 -0.5) //3
   arc 7 1 (3.5 0 -0.5) //4
   arc 1 3 (0 3.5 -0.5) //5
   arc 3 5 (-3.5 0 -0.5) //6
   arc 5 7 (0 -3.5 -0.5) //7
   arc 26 20 (0.25 0 0.5) //8
   arc 20 22 (0 0.25 0.5) //9
   arc 22 24 (-0.25 0 0.5) //10
   arc 24 26 (0 -0.25 0.5) //11
   arc 27 21 (3.5 0 0.5) //12
   arc 21 23 (0 3.5 0.5) //13
   arc 23 25 (-3.5 0 0.5) //14
   arc 25 27 (0 -3.5 0.5) //15
);

patches
(
    patch outlet
    (
        (8 9 29 28)
        (9 10 30 29)
        (10 11 31 30)
    )
    patch inlet
    (
        (14 15 35 34)
        (15 16 36 35)
       (16 17 37 36)
    )
    wall upperWall
    (
        (11 12 32 31)
        (12 13 33 32)
        (13 14 34 33)
```

```
    )

    wall lowerWall
    (
        (17 18 38 37)
        (18 19 39 38)
        (19 8 28 39)
    )
    wall cylinder
    (
       (6 0 20 26)
       (0 2 22 20)
       (2 4 24 22)
       (4 6 26 24)
    )
    empty back
    (
        (0 1 7 6)
        (3 1 0 2)
        (3 2 4 5)
        (4 6 7 5)
        (1 10 9 7)
        (12 11 10 1)
        (13 12 1 3)
        (14 13 3 15)
        (15 3 5 16)
        (16 5 18 17)
       (5 7 19 18)
       (7 9 8 19)
    )
    empty front
    (
        (21 20 26 27)
        (21 23 22 20)
        (22 23 25 24)
        (26 24 25 27)
        (30 21 27 29)
        (31 32 21 30)
        (32 33 23 21)
        (33 34 35 23)
        (23 35 36 25)
        (25 36 37 38)
       (27 25 38 39)
       (29 27 39 28)
    )
);

mergePatchPairs
(
);

// ************************************************************************* //
```

A.2. DynamicMeshDict

```
/*---------------------------------*- C++ -*----------------------------------*\
| ==      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      dynamicMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dynamicFvMesh      dynamicMotionSolverFvMesh;

motionSolverLibs (
                    "libfvM"tionSolvers.so"
                    "l"bsix"oFRigidBodyMotion.so"
             );
"solver            displacementLaplacian;

displacementLaplacianCoeffs
{
    diffusivity       inverseDistance (cylinder);
}

// ************************************************************************* //
```

Appendix B - Time Directories

B.1. Velocity (U)

```
/*---------------------------------*- C++ -*----------------------------------*\
| =====   /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      http://www.openfoam.org               |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "0";
    o“j”ct U;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform (1 0 0);
    }
    outlet
    {
        type            zeroGradient;
    }
    upperWall
    {
        type            slip;
    }
    lowerWall
    {
        type            slip;
    }
    cylinder
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
    front
    {
        type            empty;
    }
    back
    {
        type            empty;
    }
}
// ************************************************************************* //
```

B.2. Pressure (p)

```
/*--------------------------------*- C++ -*----------------------------------*\
| ========/  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      http://www.openfoam.org               |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    obje‟t”p;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }
    outlet
    {
        type            fixedValue;
        value           uniform 0;
    }
    upperWall
    {
        type            slip;
    }
    lowerWall
    {
        type            slip;
    }
    cylinder
    {
        type            zeroGradient;
    }
    front
    {
        type            empty;
    }
    back
    {
        type            empty;
    }
}


// ************************************************************************* //
```

B.3. Turbulent viscosity (`nut`) for Re = 10,000

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object " "  nut;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions      [0 2 -1 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            calculated;
        value           uniform 0;
    }
    outlet
    {
        type            calculated;
        value           uniform 0;
    }
    upperWall
    {
        type            calculated;
        value           uniform 0;
    }
    lowerWall
    {
        type            calculated;
        value           uniform 0;
    }
    cylinder
    {
        type            nutkWallFunction;
        Cmu             0.09;
        kappa           0.41;
        E               9.8;
        value           uniform 0;
    }
    front
    {
        type            empty;
    }
    back
    {
        type            empty;
    }
}
// ************************************************************************
```

B.4. Turbulent Intensity (*k*) for Re = 10,000

```
/*--------------------------------*- C++ -*----------------------------------*\
|       o          |                                                         |
|    o     o       | HELYX-OS                                                 |
|   o   O   o      | Version: v2.1.1                                          |
|    o     o       | Web:     http://www.engys.com                           |
|       o          |                                                         |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    location "0";
    object ";"}
dimensions [ 0 2 -2 0 0 0 0 ];
internalField uniform 1.512e-8;
boundaryField
{
    inlet
    {
        type           turbulentIntensityKineticEnergyInlet;
       U               U; //name of velocity field
        intensity      0.005; //0.5% turbulence
        value          uniform 1; //placeholder
    }

    outlet
    {
        type zeroGradient;
    }

    cylinder
    {
        type           kqRWallFunction;
        value          uniform 1.512e-8;
    }

    upperWall
    {
       type slip;
    }

    lowerWall
    {
       type slip;
    }

    front
    {
       type empty;
    }

    back
    {
       type empty;
    }

}
```

B.5. Turbulent Dissipation (ω) for Re = 10,000

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========     eld         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.x                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "250";
    object  "   ”mega;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 -1 0 0 0 0];

internalField   uniform 0.013554;

boundaryField
{
    outlet
    {
        type            zeroGradient;
    }
    inlet
    {
        type            fixedValue;
        value           uniform 0.013554;
    }
    cylinder
    {
        type            omegaWallFunction;
        value           uniform 0.013554;
    }
    upperWall
    {
       type         slip;
    }
    lowerWall
    {
       type         slip;
    }
    front
    {
       type         empty;
    }
    back
    {
       type         empty;
    }
}


// ************************************************************************* //
```

B.6. *pointDisplacement (*Resonance Case)

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.2                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       pointVectorField;
    location    "0.01";
    object    " poi"tDisplacement;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 0 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    cylinder
    {
        type            sixDoFRigidBodyDisplacement;
        centreOfMass    (0 0 0);
        momentOfInertia (5.984 5.984 11.632);
        mass            180;
        orientation
        (
            1 0 0
            0 1 0
            0 0 1
        );

        velocity        (0 0 0);
        acceleration    (0 0 0);
        angularMomentum (0 0 0);
        torque          (0 0 0);
        gravity         (0 -9.81 0);
        rhoName         rhoInf;
        rhoInf          1024;
        report          on;

         restraints
        {
            S1
            {
                sixDoFRigidBodyMotionRestraint linearSpring;
                    anchor          (0 -8 0); //A1
                    refAttachmentPt (0 0 0);
                    stiffness       1750;
                    damping         25;
                    restLength      8;
            }
            S3
            {
                sixDoFRigidBodyMotionRestraint sphericalAngularDamper;
                    coeff  200;
            }
        }
```

```
        constraints
        {
        yLine
              {
               sixDoFRigidBodyMotionConstraint line;
               centreOfRotation    (0 0 0);
               direction           (0 1 0);
              }
        }

        value           uniform (0 0 0);
    }

    back
    {
        type            empty;
    }

    front
    {
        type            empty;
    }

     ".*"
     {
         t"pe"           fixedValue;
         value           uniform (0 0 0);
     }
}


// ************************************************************************* //
```

Appendix C - *controlDict*

C.1. Laminar Flow

```
/*--------------------------------------------------------------------------*\
| =========                 | OpenFOAM: The Open Source CFD Toolbox          |
|  \\    /   O peration     | Version:  1.6                                  |
|   \\  /    A nd           | Web:      http://www.openfoam.org              |
|    \\/     M anipulation  |                                                |
\*--------------------------------------------------------------------------*/

FoamFile
{
    version         2.0;
    format          ascii;
    class           dictionary;
    location        "system";
    object      "    co"trolDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application potentialFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime 60;

deltaT          0.002;

writeControl    adjustableRunTime;

writeInterval   0.05;

purgeWrite      0;

writeFormat     ascii;
```

```
writePrecision  6;

writeCompression uncompressed;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;

functions
{
    forceCoeffs
    {
       type          forceCoeffs; // compute force-coefficients
       functionObjectLibs ( "libforces.so" );
       patche"       (cylinder)"
       directForceDesnity no;
       outputControl       timeStep;
       outputInterval      1;
       pName p;
       UName U;
       rhoName       rhoInf;
       log           true;
       rhoInf              1025;
       liftDir       (0 1 0);
       dragDir       (1 0 0);
       CofR          (0 0.25 0);
       pitchAxis     (0 0 0);
       magUInf       1; //magnitude of U infinity
       lRef          0.5; // reference length
       Aref          0.5; // reference area
    }
}

// ************************************************************************* //
```

C.2. Turbulent Flow (Re=10,000)

Asdaasd

```
/*---------------------------------------------------------------------------*\
| =========                 | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      http://www.openfoam.org               |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

FoamFile
{
    version         2.0;
    format          ascii;
    class           dictionary;
    location        "system";
    object        " contr"lDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application pisoFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime 3000;

deltaT          0.05;

writeControl    adjustableRunTime;

writeInterval   1;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;
```

```
writeCompression uncompressed;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;

functions
{
    forceCoeffs
    {
       type          forceCoeffs; // compute force-coefficients
       functionObjectLibs ( "libforces.so" );
       patches       “cylinder);
       ”irectForceDesnity no;
       outputControl       timeStep;
       outputInterval      1;
       pName p;
       UName U;
       rhoName       rhoInf;
       log           true;
       rhoInf              1025;
       liftDir       (0 1 0);
       dragDir       (1 0 0);
       CofR          (0 0.25 0);
       pitchAxis     (0 0 0);
       magUInf       0.02008; //magnitude of U infinity
       lRef          0.5; // reference length
       Aref          0.5; // reference area
    }
}

// ************************************************************************* //
```

C.3. Vortex Induced Vibration

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      contr“lDict;”}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     pimpleDyMFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime 90;

deltaT          0.001; //1e-04

writeControl    adjustableRunTime;

writeInterval   0.05;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression uncompressed;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;


functions
{
    forceCoeffs
    {
       type          forceCoeffs; // compute force-coefficients
       functionObjectLibs ( "libforces.so" );
       patches       (cy“inder);
       //d”rectForceDensity no;
       outputControl       timeStep;
       outputInterval      1;
       pName p;
       UName U;
       rhoName       rhoInf;
       log           true;
       rhoInf              102.4;
```

```
        liftDir       (0 1 0);
        dragDir       (1 0 0);
        CofR          (0 0.25 0);
        pitchAxis     (0 0 0);
        magUInf       1; //magnitude of U infinity
        lRef          0.5; // reference length
        Aref          0.5; // reference area
    }


}


// ************************************************************************* //
```

Appendix D - *fvScheme*

D.1. Laminar Flow (Re = 100)

```
/*--------------------------------------------------------------------------*\
| =========                 |  penFOAM: The Open Source CFD Toolbox          |
|  \\    /   O peration     | Version:  1.6                                  |
|   \\  /    A nd           | Web:      http://www.openfoam.org              |
|    \\/     M anipulation  |                                                |
\*--------------------------------------------------------------------------*/

FoamFile
{
    version         2.0;
    format          ascii;
    class           dictionary;
    location        "system";
    object          fvSc"emes;
”

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default backward;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      Gauss limitedLinear 0.5;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear limited 0.5;
    laplacian((1|A(U)),p) Gauss linear limited 0.5;
    laplacian(DkEff,k) Gauss linear corrected;*/
    laplacian(DnuTildaEff,nuTilda) Gauss linear limited 0.5;
    laplacian(nu,U) Gauss linear limited 0.5;
    laplacian(1,p) Gauss linear limited 0.5;*/
}

interpolationSchemes
{
    default         linear;
    interpolate(U)  linear;
}

snGradSchemes
{
    default         limited 0.5;
}
```

```
fluxRequired
{
    default         no;
    pcorr;
    p;
}


// ************************************************************************* //
```

D.2. Turbulent Flow (Re = 10,000)

```
/*--------------------------------------------------------------------------*\
| =========                 |     FOAM: The Open Source CFD Toolbox          |
|  \\    /   O peration     | Version:  1.6                                  |
|   \\  /    A nd           | Web:      http://www.openfoam.org              |
|    \\/     M anipulation  |                                                |
\*--------------------------------------------------------------------------*/

FoamFile
{
    version         2.0;
    format          ascii;
    class           dictionary;
    location        "system";
    object          fvSchem"s;
}

"/ * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default backward;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      Gauss limitedLinear 0.5;
    div(phi,k)      Gauss limitedLinear 0.5;
    div(phi,omega)  Gauss limitedLinear 0.5;
    div((nuEff*dev(T(grad(U))))) Gauss linear;

}

laplacianSchemes
{
    default         none;
    laplacian(nuEff,U) Gauss linear limited 0.5;
    laplacian((1|A(U)),p) Gauss linear limited 0.5;
    laplacian(DkEff,k) Gauss linear corrected;
```

```
    laplacian(DomegaEff,omega) Gauss linear corrected;
    laplacian(DnuTildaEff,nuTilda) Gauss linear limited 0.5;
    laplacian(nu,U) Gauss linear limited 0.5;
    laplacian(1,p) Gauss linear limited 0.5;
}

interpolationSchemes
{
    default         linear;
    interpolate(U)  linear;
}

snGradSchemes
{
    default         limited 0.5;
}

fluxRequired
{
    default         no;
    pcorr;
    p;
}


// ************************************************************************* //
```

D.3. Vortex Induced Vibration

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |        M: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.2                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default Euler;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}

divSchemes
{
    default         none;
```

```
    div(phi,U)      Gauss linearUpwind grad(U);
    div(phi,k)      Gauss limitedLinear 1;
    div(phi,omega)  Gauss limitedLinear 1;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear limited corrected 0.5;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         uncorrected;
}

fluxRequired
{
    default         no;
    pcorr           ;
    p;
}

// ************************************************************************* //
```

Appendix E - *fvSolution*

E.1. Laminar Flow

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |           The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// " * * *"* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-06;
        relTol          0.1;
        smoother        GaussSeidel;
        nPreSweeps      0;
        nPostSweeps     2;
        cacheAgglomeration on;
        agglomerator    faceAreaPair;
        nCellsInCoarsestLevel 10;
        mergeLevels     1;
    }

    pFinal
    {
        $p;
        tolerance       1e-06;
        relTol          0;
    }

    "(U|k|epsilon|R|nuTilda)"
    {
        sol"er          smoothSolve";
        smoother        GaussSeidel;
        tolerance       1e-05;
        relTol          0;
    }
}

potentialFlow
{
    nNonOrthogonalCorrectors 20;
}

PISO
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 1;
```

```
    pRefCell        0;
    pRefValue       0;
}


// ************************************************************************* //
```

E.2. Turbulent Flow

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                 Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                        |
|    \\/     M anipulation  |                                                   |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * *"* * * " * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-06;
        relTol          0.1;
        smoother        GaussSeidel;
        nPreSweeps      0;
        nPostSweeps     2;
        cacheAgglomeration on;
        agglomerator    faceAreaPair;
        nCellsInCoarsestLevel 10;
        mergeLevels     1;
    }

    pFinal
    {
        $p;
        tolerance       1e-06;
        relTol          0;
    }

    "(U|k|omega|R|nuTilda)"
    {
        solver  "       smoothSolver;
"       smoother        GaussSeidel;
        tolerance       1e-05;
        relTol          0;
    }
}

potentialFlow
{
    nNonOrthogonalCcorrectors 20;
}

PISO
```

```
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 1;
    pRefCell        0;
    pRefValue       0;
}


// ************************************************************************* //
```

E.3. Vortex Induced Vibration

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                 en Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.2                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/ ___ M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    pcorr
    {
        solver           GAMG;
        tolerance        0.02;
        relTol           0;
        smoother         GaussSeidel;
        nPreSweeps       0;
        nPostSweeps      2;
        cacheAgglomeration on;
        agglomerator     faceAreaPair;
        nCellsInCoarsestLevel 10;
        mergeLevels      1;
    }

    p
    {
        $pcorr
        tolerance        1e-7;
        relTol           0.01;
    }

    pFinal
    {
        $p;
        tolerance        1e-7;
        relTol           0;
    }

    "(U|k|omega)"
    {
        solver          PBiCG"
        pr"conditioner  DILU;
        tolerance       1e-06;
        relTol          0.1;
```

```
    }

    "(U|k|omega)Final"
    {
        $U;
        tole‟ance       1e-06”
        relTol          0;
    }

    cellDisplacement
    {
        solver          GAMG;
        tolerance       1e-5;
        relTol          0;
        smoother        GaussSeidel;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }
}

PIMPLE
{
    correctPhi          yes;
    nOuterCorrectors    2;
    nCorrectors         1;
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    fields
    {
        p               0.3;
    }
    equations
    {
        "(U|k|omega)"   0.7;
        "(U|k|omega)Final" 1‟0;
    }
}
”cache
{
    grad‟U);
}

// ************************************************************************* //
```

Appendix F - Troubleshooting the Resonance Case

Problem: large vertical displacement during resonance.

- Solution 1: Increase the domain size until it is large enough to capture the movement of the pipe. If the domain is too small, the mesh will be displaced as illustrated in Figure G.1.

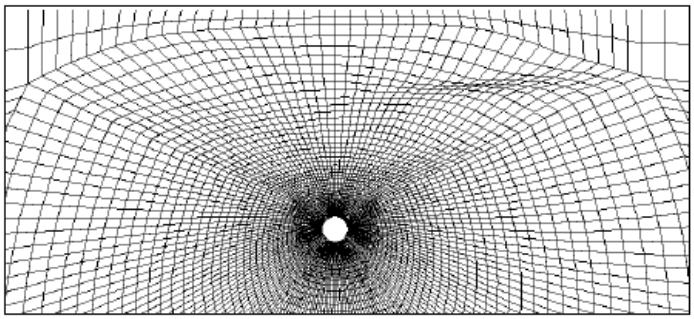


Figure G.1: Displaced dynamic mesh when the domain is too small to capture the large displacement of the pipe

Solution 2: Apply damping on the vertical spring. Damping will decrease the amplitude of the motion but will not affect the oscillating frequencies at all, as described in Figure G.2.

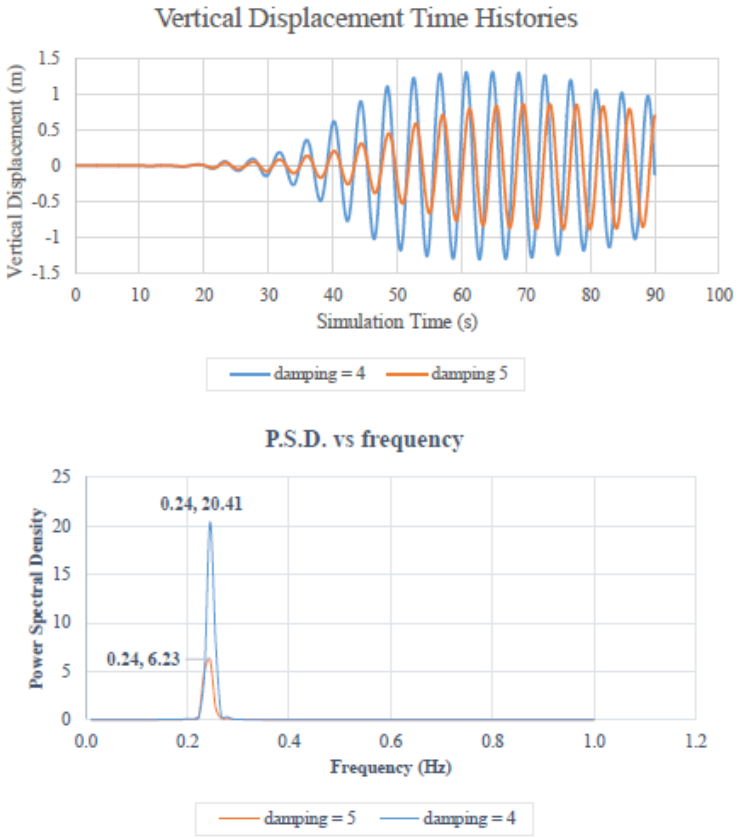


Figure G.2: Displacement histories and spectral density comparison for cases with different damping

Problem: roll motion of the pipe, as illustrated in Figure G.3.

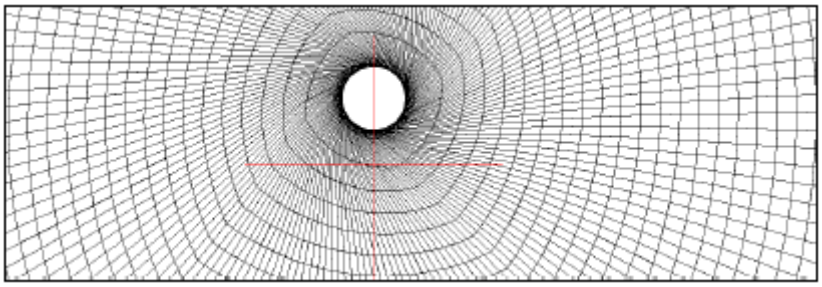


Figure G.3: Roll motion of the pipe

- Solution 1: apply angular damper with a very high value of damping
- Solution 2: increase the value of the moment of inertia in z-direction until it is close to infinity.

Appendix G - Content of Enclosed CD

The enclosed CS consists of the master thesis files such as follows:

- Thesis report in pdf format
- Case setup folders (0, constant, and system directories) of:
    - laminar flow simulation (11 cases)
    - turbulent flow simulation (7 cases)
    - pipe with seabed proximity simulation (4 cases)
    - vortex induced vibration of a pipe with 1 DoF (5 cases)
- Animation video of velocity profile in paraView of:
    - laminar flow simulation (11 cases)
    - turbulent flow simulation (7 cases)
    - pipe with seabed proximity simulation (4 cases)
    - vortex induced vibration of a pipe with 1 DoF (5 cases)