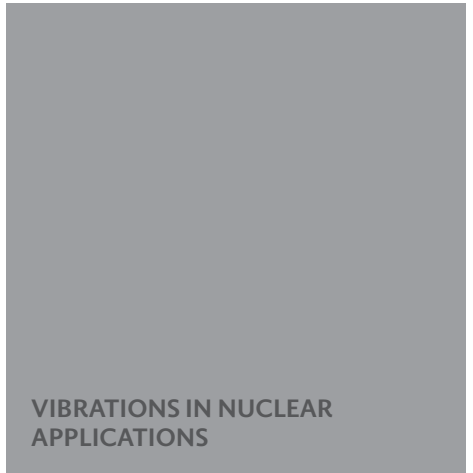
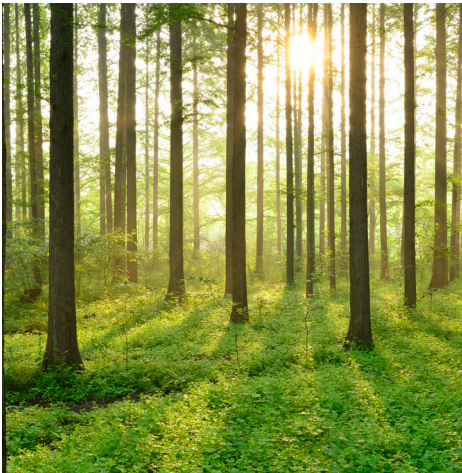


HPC FSI SIMULATION OF A ROD SUBJECTED TO AXIAL TURBULENT FLOW

REPORT 2016:269



HPC FSI simulation of a rod subjected to axial turbulent flow

Calculations for nuclear power applications

VASILEIOS PAPADIMITRIOU

ISBN 978-91-7673-269-4 | © 2016 ENERGIFORSK

Energiforsk AB | Phone: 08-677 25 30 | E-mail: kontakt@energiforsk.se | www.energiforsk.se

Foreword

In a previous project, experiments have been conducted in a set up with a geometry equivalent to the neutron detector housing tube and consists of four fuel box corners and the housing tube. The geometry used has then been implemented into a fluid structure interaction (FSI) software, to investigate if the calculated and experimental results are consistent.

The FSI calculations have been carried out as a master thesis project at The Royal Institute of Technology/Stockholm University in cooperation Vattenfall R&D. In this thesis work, an open source software was used.

In a parallel master thesis, performed at Chalmers/Onsala Ingenjörskyrå (Energiforsk report 2016:238), the same modelling problem is studied using a commercial software was used. The results using the two softwares will also be compared both to each other and to the experimental results in a separate summarizing report.

This project has been carried out within the Energiforsk Vibrations research program. The stakeholders of the Vibrations program are Vattenfall, E.ON, Fortum, TVO, Skellefteå Kraft and Karlstads Energi.

HPC simulation of fluid-structure interaction for an in-core neutron flux detector housing tube

Vasileios Papadimitriou

Department of Physics

Degree project 30 HE credits

Degree subject: Computational Physics

Degree project at the Master level

Autumn 2015

Advisor: Eva Lindroth

Assistant advisor: Lilit Axner

External advisor: Eric Lillberg

Reviewer:

Swedish title:



Stockholm
University

HPC simulation of fluid-structure interaction for an in-core neutron flux detector housing tube

Vasileios Papadimitriou

Abstract

The study of flow-induced vibrations on slender structures is a relatively new subject, with increasing interest in recent years. In the present work the axial turbulent flow past a cylindrical object is simulated, by using an open source CFD code (OpenFOAM) with a Large Eddy Simulation (LES) approach. High Performance Computing (HPC) resources were available and a detailed approach has been accomplished, with several computational meshes scaling up to 20 million cells. Spectral and spatial analysis has been performed on the numerical results and validation with the experimental data provided (by the projects contributors) has been carried out. Additionally, an FSI simulation has been achieved for the same geometry, with a partitioned approach, coupling two separate flow and structural solvers and a comparison has been performed with the experiment here as well, in order to examine the ability of the software to predict the behavior of the FSI and how the numerical approximations affect the results in turbulent cases. The turbulent velocity frequencies showed a good correspondence with the natural vibration modes of the rod and the software was shown capable to simulate the FSI behavior, but had room for improvement in several sectors, especially HPC.

Keywords

Turbulence, Fluid-Structure Interaction, Large Eddy Simulation, High Performance Computing, Cylindrical slender structure, Boiling Water Reactor, OpenFOAM, scalability, performance

Contents

1	Introduction	11
1.1	Problem Description	11
1.2	Objective of this study	13
1.3	Materials used for this thesis	13
1.4	Acknowledgements	13
1.5	Thesis outline	14
2	Background	15
2.1	The Finite Volume Method	15
2.2	The governing equations of fluid flow	15
2.2.1	Navier-Stokes equations for a Newtonian fluid	16
2.3	Turbulence	16
2.3.1	Turbulence Models	17
2.4	Fluid Structure Interaction	19
2.4.1	Monolithic and Partitioned Solvers	19
2.4.2	Definition of the flow and structural solvers	19
2.4.3	Weak and strong coupling	20
2.4.4	Fixed point iterations (sub-iterations)	20
2.4.5	Stability - Under-relaxation	21
2.4.6	Aitken under-relaxation	21
2.4.7	IQN-ILS	21
2.5	ALE formulation	23
2.6	Motion for the bending vibration of a slender homogeneous beam	24
3	Implementation	25
3.1	Meshing	25
3.2	Turbulence	26
3.2.1	Turbulence with OpenFOAM	27
3.3	Sampling	30
3.3.1	Experimental setup sampling	30
3.3.2	Sampling with OpenFOAM	30
3.3.3	Field Averages	31
3.3.4	Probe sampling	31
3.4	Fluid Structure Interaction	33
3.4.1	FSI with OpenFOAM	33
3.4.2	Fundamental frequencies of structural domain	35
4	Results & Discussion	39
4.1	Turbulence	39
4.1.1	RMS values and variances	39
4.1.2	Spectral Analysis	43
4.1.3	Matrix solver performance	46
4.2	FSI simulation	48
4.2.1	Performance	50
4.2.2	Mesh Displacement	50
4.2.3	Spectral Analysis	51

5	Future work & Conclusion	59
----------	-------------------------------------	-----------

List of Figures

1.1	Experimental geometry.	12
2.1	Example of turbulent velocity over time.	17
3.1	Computational mesh in flow direction.	26
3.2	Computational mesh as seen from the outside.	26
3.3	Computational mesh as seen from the inside.	27
3.4	PISO algorithm flow chart.	28
3.5	Location of pressure transducers in experimental setup.	31
3.6	Line locations for RMS and variance plotting.	32
3.7	Locations of probes inside the computational domain.	33
3.8	<i>FsiFoam</i> algorithm flow chart.	34
4.1	Magnitude of the velocity inside the computational domain.	40
4.2	Velocity profile in flow direction.	40
4.3	Plots over lines for velocity magnitude, mean and variance for channel flow at 900mm height.	42
4.4	Frequency spectra for turbulent velocity's x component.	45
4.5	Energy spectrum.	47
4.6	Time over cells/core for different matrix solvers.	49
4.7	Displacement of a point on the structural mesh in a surface vertical to the flow normal direction.	52
4.8	Displacement of a point on the structural mesh over time in the flow normal direction.	52
4.9	Spectrum of the housing tube's x and y velocities at a height of 350mm. . .	53
4.10	Spectrum of the housing tube's x and y velocities at a height of 550mm. . .	54
4.11	Spectrum of the housing tube's x and y velocities at a height of 750mm. . .	54
4.12	Spectrum of the housing tube's x and y velocities at a height of 950mm. . .	55
4.13	Spectrum of the housing tube's x and y velocities at a height of 1150mm. .	55
4.14	Spectrum of the housing tube's x axis displacement at a height of 950mm. .	56
4.15	Spectrum of the pressure along the tube's surface for the x component. . .	56
4.16	Frequency spectra for experimental setup (provided by the project's supervisor[1]).	57

List of Abbreviations

ALE	Arbitrary Lagrangian-Eulerian formulation
APRM	Average Power Range Monitoring system
CFD	Computational Fluid Dynamics
CV	Control Volume
DES	Detached Eddy Simulation
DNS	Direct Numerical Simulation
FIV	Flow Induced Vibrations
FOAM	Field Operation and Manipulation
FSI	Fluid-Structure Interaction
FSO	Full Scale Output
FV	Finite Volume
GGI	General Grid Interface
HPC	High Performance Computing
IQN-ILS	Interface Quasi-Newton iterations based on the Inverse Least squares approximation of the Jacobian
LES	Large Eddy Simulation
PDE	Partial Differential Equation
PISO	Pressure Implicit with Splitting of Operator algorithm
PSD	Power Spectral Density
RANS	Reynolds Averaged Navier-Stokes
RMS	Root Mean Squared
STL	Stereolithography

1. Introduction

The study of flow-induced vibrations of slender structures subjected to axial flow is a relatively new subject, the study of which has been gradually intensified in the past few decades. The ever growing power generating industry, as well as other engineering industrial fields, has found itself in need of stability and equipment reliability. Repeated equipment failures, that have the possibility to cause unwanted effects with a severe impact, sometimes, in their natural habitat, have evidenced the inadequate state of the art repeatedly [2]. It has thus become important to understand and predict the dynamical behavior of such structures, which can be found in many mechanical equipment, a few primary examples being nuclear reactors [3] [4], steam generators and heat exchangers. Most failures are associated with cases of cross-flow, however lately, vibrations due to axial flow have been shown to be of great importance as well [5]. It is expected that the dynamical behavior of slender structures subjected to external axial flow should be different than slender structures conveying fluid or being subjected to cross-flow [6]. With this as a motivation, the current study attempts to analyze and understand the dynamic behavior of a beam subjected to axial external flow, using as a base existing studies in similar cases, with an emphasis on industrial applications.

1.1 Problem Description

This study focuses on the simulation of the external axial flow surrounding a slender cylindrical structure and the vibrations induced due to the interaction between the structure and the fluid flow. The geometry considered in the experimental setup is a part of a nuclear in-core neutron flux monitoring system. This system, which is called the Average Power Range Monitoring (APRM) system, consists of several in-core neutron flux detectors that are positioned axially inside guide tubes that are located in between the fuel bundles of the reactor [1]. Approximately 13-15% of the total flow of the water in the reactors main recirculation loop flows axially in the cross shaped channel formed between four fuel bundles. In the middle of that channel, the guide tube containing the neutron flux detectors is located and is subjected to this flow on its outer surface. Due to the guide tubes structure, which is only 19 mm in diameter but 4040 mm long, it is susceptible to deformation due to the fluid forces, and thus a Fluid Structure Interaction (FSI) is created. The bottom end of the guide tube is fixed in the core support plate while the top end is supported on the core grid with a spring. This gives the system a fundamental frequency of 2-3 Hz.

The experimental geometry, that is simulated, had to be scaled down due to the limited length of the available test section and can be seen on figure 1.1 [1]. The length of the test tube is only 1500 mm long, made of stainless steel and has a diameter of 8mm. In order to scale the whole geometry to match the characteristics of the guide tube, a channel width between the fuel bundles of 6.4 mm and rounded corners with a corner radius of 5.4 mm was obtained, leaving a circular area of 13.4 mm for the test tube. Thus the available flow channel between the guide tube and the fuel bundles is a small annular passage of 5.4 mm in diameter. The inflow region is defined by the corners of the fuel bundles with a mass flow ranging from 0 to 15 m^3/s . The outflow boundary and piping is connected to a reservoir at constant pressure. The mechanical properties of the geometry have been summarized on table 1.1.

Property	Value
Rod Length	1.486m
Rod diameter	0.008m
Rod wall thickness	0.0006m
Rod density	7863 kg/m^3
Young's modulus	193 GPa
Water density	998 kg/m^3

Table 1.1: *Mechanical properties of experimental geometry.*

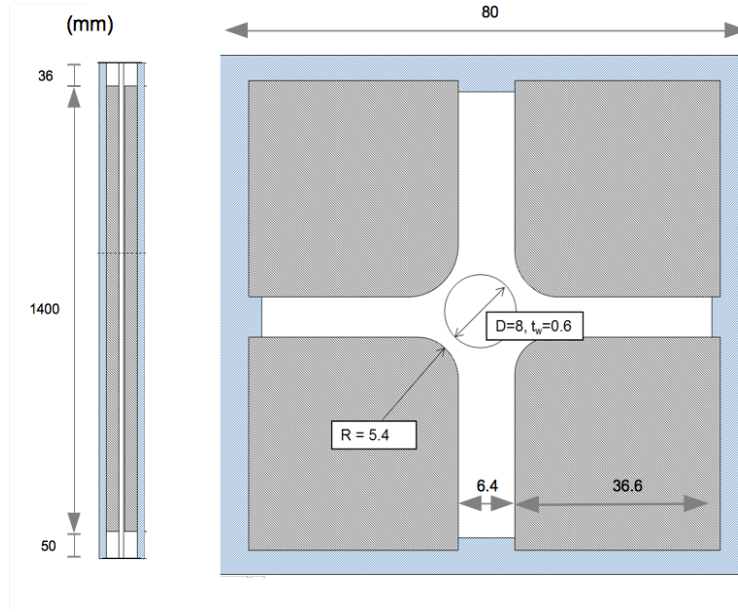


Figure 1.1: Experimental geometry.

The main issue with this setup is the fact that the tube in question is susceptible to FIV due to the flow surrounding it and its effects and causes have to be well understood and controlled in order to be allowed to exist.

According to studies concerning FIV in cylindrical structures subjected to axial flow [7] and reactor components specifically [3], there are 3 reasons for such vibrations to occur. First, fluid elastic instabilities like buckling and fluttering are said to occur when there is a high velocity flow field ($u^+ > 1.5$). Next, vibrations due to flow periodicity are possible, if there are sizable perturbations in the mean velocity flow field which can cause parametric resonances if that perturbation frequency lies in the neighborhood $2f_n/k$, $k=1,2,\dots$. Finally, FIV due to turbulent buffeting is a possibility, where the cylinder acts as a band-absorption filter, extracting energies from the flow field in narrow windows about the natural frequencies of the cylinder, especially the first mode.

1.2 Objective of this study

The objectives of this study have been set as follows:

- Investigation of the turbulent flow characteristics through the geometry of interest, without FSI coupling at first, using a Large Eddy Simulation (LES) turbulence model, to understand the inner workings of the flow, study its cause and effects on the structure as a whole and try to understand the cause of the Flow Induced Vibrations (FIV) of the guide tube using a spectral analysis on the simulation basic components and comparing with experimental data provided.
- Strongly coupled simulation of the FSI between fluid and solid domain, benchmarking of the solvers with comparison to the experimental geometry and the effectiveness of *OpenFOAM* as a software capable of industrial FSI applications and High Performance Computing (HPC) simulations.

1.3 Materials used for this thesis

The software used for this study is called *OpenFOAM* (Field Operation and Manipulation). *OpenFOAM* is a free, open-source Computational Fluid Dynamics (CFD) software package. It is given as a set of libraries, written in C++, used to create executable files. These libraries have files with shared functionality for easier categorization. There are two groups of applications: solvers, with the purpose of handling the numerics and the solution of the mechanical problem at hand, and utilities, that are tasked with performing non-essential tasks that involve data manipulation, meshing or visualization [8]. Additionally, a community-extended version of the software is going to be used, named *foam-extend* [9] that has a solid mechanics branch with implemented structural and FSI solvers, alongside the original *OpenFOAM* release for meshing, pre/post-processing tools and a turbulence study. Furthermore, the open-source software *Salome* has been used for generation of the Computer Aided Design (CAD) model used for the mesh generation of this work [10]. Finally, the open-source software *Octave* has been used for graphical representations and data analysis. *GNU Octave* is a high-level interpreted language, with capabilities for numerical procedures and extensive graphical capabilities for data visualization and manipulation [11].

1.4 Acknowledgements

This thesis, as well as the experiments it is based on, are financed by *Energiforsk AB*. An active collaborator is *Vattenfall* and its R&D department that contributed to the realization of the experiment, as well as part of the supervision of this study and provision of a detailed report to the experimental data and structure, which acted as a basis for this study. The thesis itself was conducted at the *PDC* center for High Performance Computing at the KTH Royal Institute of Technology [12]. *PDC* provides HPC services to Swedish academia, as part of the *Swedish National Infrastructure for Computing (SNIC)* as well as internationally via *PRACE* infrastructure. Their computer systems have been used for the computations, specifically the cluster *Beskow*. *Beskow* is a Cray XC40 system with 1676 compute nodes and 26 service nodes. Each node consists of 32 processing units (cores) and 64GB of computing memory. It has a peak performance of 1973 TFLOPS (floating points per second) and was ranked in the 32nd position in the November 2014 Top 500 list worldwide.

1.5 Thesis outline

The rest of this report is structured as following:

- Chapter 2 provides the background information and any previous knowledge required to proceed on understanding the content of this study. First it briefly describes CFD and the equations describing the fluid flow. Then it proceeds to describe the idea of turbulence and its characteristics and finally delves into FSI and its required theoretical background.
- Chapter 3 explains the implementation of a simulation in OpenFOAM, both in the turbulent and the FSI case. It provides the setup as well as the specific properties defined (boundary and initial conditions, constant values) and finally dives into the strategy for obtaining and analyzing the data from these simulations.
- Chapter 4 provides the results obtained from the study with tables and graphical representations along with a brief explanation of them. It also lists the reasons they were collected, as well as any necessary information for their interpretation and a discussion of their meaning and importance.
- Finally, chapter 5 provides a conclusion for this work and discusses any future work that could be carried out on the same subject, along with final thoughts and entries that did not fit in the rest of the report.

2. Background

Computational Fluid Dynamics (CFD) is a discipline that belongs to the field of mechanics that involves using numerical methods and computational schemes to obtain approximate solutions to the complicated Navier - Stokes equations, that describe fluid motion. These equations, in order to be solved, have to be discretized by some method (for example with finite volumes or finite elements) for the specific geometry of interest, which represents an approximation for the differential equations by a system of algebraic equations. Using this system to solve for the fluid motion equations in space and/or time, you can create a numerical solution and thus, provide results at distinct (and small) locations in space and/or time [13].

2.1 The Finite Volume Method

In the Finite Volume method, the computational domain is split into very small cubical segments, called control volumes (CVs) to form a mesh surrounding the computational nodes instead of being defined by them. These are considered to be either at the center of the control volumes (first approach) or be considered first, with the control volumes constructed afterwards around them, in such a way that the CV faces lie midway between nodes (second approach). The first approach has the advantage that the values of different quantities on the CV represent the mean of that quantity on the CV to a higher accuracy, since the node is located exactly at the center of the CV. On the other hand, the second approach provides higher accuracy during linear interpolation of values on the CV faces since the face lies midway between two nodes [13].

2.2 The governing equations of fluid flow

The governing equations of fluid flow are called the Navier-Stokes equations. These were derived in the mid 18th century by Claude Navier and George Stokes and are based on the three fundamental governing equations of fluid dynamics. These equations are called the continuity, momentum and energy equations which represent the conservation laws of physics. The continuity or mass-conservation equation demands that the "rate of increase of mass in a fluid element" is equal to the "net rate of mass flow into the fluid element" [14]. The rate of increase of mass in a fluid element is [14]

$$\frac{\partial}{\partial t}(\rho \delta x \delta y \delta z) = \frac{\partial \rho}{\partial t}(\delta x \delta y \delta z) \quad (2.1)$$

Here, ρ refers to the fluid density field, x,y and z correspond to a Cartesian coordinate system and t is the time variable.

Furthermore, considering the CVs, this simplifies to :

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (2.2)$$

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho u) = 0 \quad (2.3)$$

This is the "three dimensional, unsteady continuity equation for a point in a compressible fluid" [14]. For incompressible fluids, the density ρ is constant, so the time derivative in respect to the density is zero and thus [14]

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (2.4)$$

is all that's needed for the continuity equation.

The momentum equation, which is based on Newton's law of motion, correlates the acceleration of the fluid's unit mass to the forces that act on the fluid. The rate of increase of momentum on a unit mass should be equal to the summation of the forces on said mass [15]. The rate of increase of momentum per unit volume is given by $\rho \frac{\partial u_i}{\partial t}$ while the sum of the forces is given by the derivative of the stress tensor t_{ij} for the surface forces, which are of molecular origin and symmetric, and gravity. Given the gravitational potential Ψ , the body force per unit mass is $g = -div(\Psi)$ [15]. Putting all these together in one equation, we derive the momentum equation [15]

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial \rho}{\partial x_i} + \frac{\partial t_{ij}}{\partial x_j} - \rho \frac{\partial \Psi}{\partial x_j} \quad (2.5)$$

2.2.1 Navier-Stokes equations for a Newtonian fluid

To obtain a more suitable form of the momentum equations, we introduce a suitable model for the stress tensor t_{ij} [15]. This is achieved by considering a specific class of fluids, called "constant property Newtonian fluids" [15]. As stated by the Boussinesq hypothesis, in most fluid flows the viscous turbulent stresses are functions of the mean deformation rate or strain rate and in Newtonian fluids they should be linearly proportional to each other. When working in three dimensions, this rate can be distinguished in the linear and volumetric deformation rate [14].

Newtonian fluids are the simplest mathematical models which account for viscosity and while no fluids are perfectly Newtonian, the most common ones, water and air exhibit Newtonian behavior under ordinary conditions. In this case the stress tensor is given by [15]

$$t_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.6)$$

where μ is the (constant) coefficient of viscosity.

By substituting this expression for the stress tensor into the general momentum equation derived earlier and using the fact that ρ and μ are uniform as well as the continuity equation $div(u) = 0$, we obtain the Navier-Stokes equations (the gravitational forces are ignored due to their low significance) [15]

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) \quad (2.7)$$

2.3 Turbulence

Turbulence is a property in fluid flow where, unlike the normal steady flow, chaotic effects take place, specifically rapid fluctuations in velocity and pressure in space and time [15]. For instance, measurement of the velocity component v at some stationary point in the

flow may produce a plot as shown in the figure 2.1. The velocity can be regarded as consisting of an average, mean, value (red line) and a randomly fluctuating value (blue line).

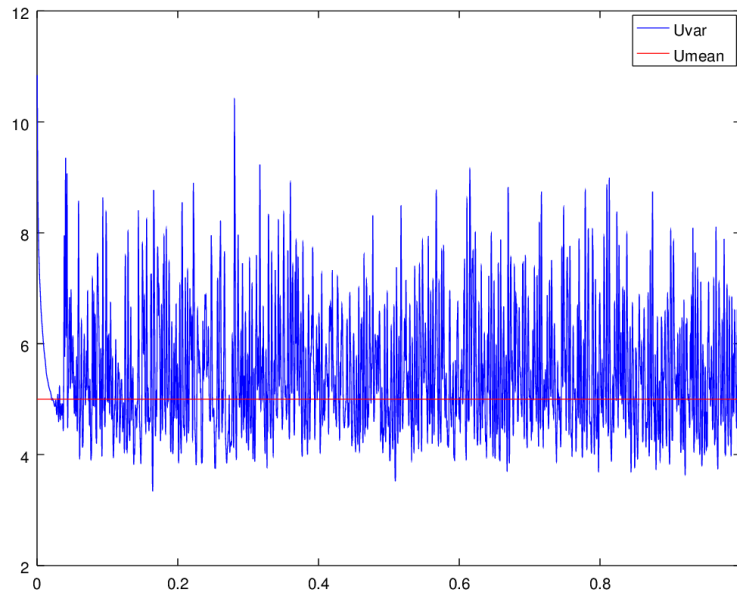


Figure 2.1: Example of turbulent velocity over time.

In turbulent flows momentum effects dominate over viscous effects and a characteristic quantity that is used as a qualitative measure of existence of turbulent effects is the Reynolds number $Re = uL/\nu$ where u and L are the characteristic velocity and length scale of the flow respectively and ν is the kinematic viscosity [15]. In pipe flow the characteristic velocity is considered the mean flow velocity through the body of the pipe and the characteristic length is the diameter of the pipe D , leading to typical values of the Reynolds number of $Re_D < 2100$ for laminar flow and $Re_D > 4000$ for turbulent flow. In the interval between these two limits, laminar and turbulent flows are both possible and are called transition flows, which depend on the geometry being examined [15]. Turbulence becomes visible in the flow field in the form of "eddies" [14][15]. There is a large range of scales for the size of the eddies where the biggest ones are in similar size to the flow geometry. Turbulent flows make the task of obtaining direct solutions to the Navier-Stokes equations extremely difficult with its wide range of lengths and sizes.

2.3.1 Turbulence Models

Turbulent flows are characterized by fields that vary highly over time both in time and space. Since these variations occur in both the large and small scales of the flow, it is computationally expensive to directly simulate everything for more complex problems that have practical engineering significance. There are two methods commonly used to eliminate this need of direct simulation of every different scale of the numerical problem, filtering and time averaging [15].

Space filtering is used to filter out the small scale fluctuations in the velocity, leaving only the large scale eddies for direct simulation, which is computationally feasible with

modern hardware. A simulation using this approach is known as a Large Eddy Simulation (LES) [13]. When the Navier-Stokes equations for an incompressible flow are filtered, one obtains the following set of equations [13]:

$$\frac{\partial(\rho \bar{u}_i)}{\partial t} + \frac{\partial(\rho \bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right] \quad (2.8)$$

and, for the continuity equation [13]:

$$\frac{\partial(\rho \bar{u}_i)}{\partial x_i} = 0 \quad (2.9)$$

Filtering the Navier-Stokes equations leaves the equations incomplete, demanding additional terms that must be explicitly modeled to acquire a closed set of equations, which are called the sub-grid scale Reynolds stresses nomenclatureSGSSub-Grid Scale (SGS) [13] :

$$t_{ij}^S = -\rho(\bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j) \quad (2.10)$$

Approximation of these SGS stresses leads to specific numerical models, the most common being the so called eddy-viscosity models. In these models, the turbulent effects are considered to be caused by an artificial quantity, called eddy-viscosity which, along with eddy-diffusivity, are quantities used to successfully model the turbulence effects, like momentum and energy transfer [13].

Equation 2.10 requires a closure for the unknown coefficients. "Following the Boussinesq approximation, the relationship between the anisotropic part of the SGS stress tensor and the strain tensor can be given as" [16] :

$$t_{ij}^S - \frac{1}{3} \delta_{ij} t_{kk}^S = -2\nu_T \bar{S}_{ij} \quad (2.11)$$

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \quad (2.12)$$

The SGS eddy-viscosity ν_T is obtained from the kinetic energy [16] :

$$\nu_T = C_k \bar{\Delta} \sqrt{k_{SGS}} \quad (2.13)$$

where the kinetic energy k_{SGS} is calculated out of the velocity field as [16] :

$$k_{SGS} = \frac{1}{2} t_{kk}^S = \frac{1}{2} (\bar{u}_k \bar{u}_k - \bar{u}_k \bar{u}_k) \quad (2.14)$$

Finally, the grid filter length $\bar{\Delta}$ is based on the cell volume [16]

$$\bar{\Delta} = (\Delta_1 \Delta_2 \Delta_3)^{\frac{1}{3}} \quad (2.15)$$

and the coefficient C_k is a flow-dependent quantity which can be computed as a scalar function of the vorticity and the resolved strain rates [16].

In time averaging all flow components are divided into a mean component and a fluctuating component which are then averaged over time to remove the effect of the fluctuation in the equations, called Reynolds averaged Navier-Stokes equations (RANS). These models should be regarded as engineering approximations rather than scientific laws and are of no further concern to this study.

2.4 Fluid Structure Interaction

Fluid Structure Interaction (abbreviated as FSI) is the interaction of some movable or deformable structure with an internal or surrounding fluid flow. Usually the structure's dislocation and/or deformation occurs due to the effect of this interaction between itself and the fluid flow which, in turn, causes the boundary of the fluid system to move. Thus, an FSI simulation is a problem of both fluid and solid mechanics. In many engineering applications, this dynamic interaction plays a key role in the efficiency and sometimes even in the safety of the design. An example of the importance of this interaction is the well-known catastrophic collapse of the Tacoma Narrows Bridge in 1940 due to an aeroelastic flutter effect from the interaction between the bridge (structure) and the airflow (fluid) around it [2].

FSI phenomena occur in many engineering applications, including reactors, aerospace applications, civil engineering as well as biomedical applications, such as arteries and artificial heart valves. Due to the advancement of computational power, as well as the booming increase of interest in research on the field and thus the advancement in the numerical algorithms available to the researchers, we are able to simulate increasingly complex problems.

2.4.1 Monolithic and Partitioned Solvers

There is a broad selection of available algorithms and methodologies to approach an FSI problem. One of the first choices that need to be made is whether to develop a monolithic or a partitioned solver [17]. For loosely coupled problems, the partitioned approach is the default choice but for strongly coupled problems and strong interaction between the two solvers a monolithic approach may be a consideration. A monolithic scheme is one that aims at putting all the necessary components needed for the simulation into a single solver. Although it has been used in the past, most researchers agree that monolithic approaches are impractical since they are not only hard to implement but keeping the solver up-to-date with the latest developments in every different research field is a challenging task [17] [18]. Therefore, the most popular approach is to develop a partitioned solver. A partitioned scheme aims at separating the multiphysics problem that the user wants to solve in two (or more) separate "black-box" solvers defined by the physical properties of the simulation. For a fluid - structure interaction, we would require a solver for the fluid, a solver for the structure and a coupling algorithm that couples the solvers at the fluid - structure interface both in space and time by transferring data between the 2 solvers by the use of some interpolation technique. All of these separate solvers can be developed, maintained and improved independently from each other, making this approach much more simple and agile.

2.4.2 Definition of the flow and structural solvers

A definition of the functions F and S that represent the flow and structural solvers respectively has to be given [19]. Because we are talking of a partitioned approach and black-box solvers, the exact Jacobians of these functions are not available. First, the discretized position $x \in \mathbb{R}^u$ of the fluid-structure interface is given to the flow solver and the grid of the fluid domain in contact with the interface is moved accordingly. Afterwards, the flow equations are solved for the whole fluid mesh which results in a distribution of stresses on the interface $y \in \mathbb{R}^w$. Thus, we have an interaction of the form $y = F(x)$. Sub-

sequently, this stress on the interface y is given to the structural solver which, in turn, solve the structural equations for the whole structural mesh and returns the new position of the fluid-structure interface x , creating an interaction of the form $y = S(y)$. Using the definitions of these two functions, the FSI problem can be represented as

$$R(x) = S \circ F(x) - x = 0 \quad (2.16)$$

with R the residual operator of the coupled problem, creating a sequence of solution of each solver and variable passing until the residual becomes small enough for the solution to be considered convergent.

2.4.3 Weak and strong coupling

As mentioned earlier, there are two forms of coupling between the fluid and solid solvers for an FSI simulation. You can either have a loosely or a strongly coupled problem [17]. When studying problems that the deformations are small and the interactions are a secondary issue, an explicit (weak) coupling method may be applied. On the other hand, for big deformations, with a primary interest in their effect on the total simulation, an explicit (strong) coupled method may be required. The main difference between these two methods is the addition of an external (outer) loop at every time step that ensures that the mesh deformation is consistent with the cinematic and dynamic conditions at the interface between the two solvers. This outer looping is called fixed-point or sub-iterating.

2.4.4 Fixed point iterations (sub-iterations)

When faced with instationary problems and moving grids, additional measures need to be taken into consideration during the development of a partitioned solver for a fully coupled problem. While in a normal algorithm you only have the main iteration, which represents the passage of time during the simulation, in fully coupled FSI problems a sub-iteration has to be set for every time step in order to achieve the strong coupling and have a convergent solution. These sub-iterations are referred to as fixed point iterations [20]. At each iteration an approximation to the exact solution is obtained. When the exact solution is not reached, a residual is introduced. Thus, this solution has to be sub-iterated until this residual is zero. Perfect convergence is not actually possible but setting a convergence criterion (a really small value), below which we consider the solution converged, is sufficient and is the common solution. Considering these fixed-point iterations as equations, the iteration can be thought as a function f of the approximate solution x ,

$$x = f(x) \quad (2.17)$$

This is then iterated until the convergence criterion is reached,

$$x^{n+1} = f(x^n) \quad (2.18)$$

Like mentioned above, when the solution x^{k+1} is not reached but only an approximation of that $\tilde{x}^{k+1}$, a residual r is introduced

$$\tilde{x}^{n+1} = x^{n+1} + r^n = f(x^n) + r^n = \tilde{x}^n + r^n \quad (2.19)$$

Equation 2.19 can be then iterated until $\tilde{x}^{n+1} = x^{n+1}$ and the residual is zero.

2.4.5 Stability - Under-relaxation

However, fixed point iterations are expensive computationally and the convergence is not guaranteed to be stable. A way to treat this instability is to apply an under-relaxation method, but at the cost of even slower convergence [20]. Under-relaxation is effective in the case of overestimation of the coupling variables. Like already explained, in fixed point iterations a new value is approximated for the solution at each sub-iteration and a residual is introduced. When under-relaxation is applied the new input variable is a combination of the approximate solution and the previous solution.

$$x^{n+1} = \theta \tilde{x}^{n+1} + (1 - \theta)x^n \quad (2.20)$$

The combination of these two variables depends on the under-relaxation factor θ . This parameter takes values in the interval $[0, 1]$ with $\theta = 1$ representing the approximation without applying any under-relaxation at all and $\theta = 0$ giving the old solution, thus not allowing the process to progress at all (stationary solution).

2.4.6 Aitken under-relaxation

If the under-relaxation parameter remains fixed during the iterations, a fixed under-relaxation scheme is achieved. It can, however, change between each sub-iteration, which results in an adaptive under-relaxation scheme. A popular adaptive under-relaxation scheme is the Aitken under-relaxation [21]. The Aitken method varies the under-relaxation parameter between each sub-iteration. For the first iteration a fixed under-relaxation factor θ is chosen. Then for future iterations this parameter is changed based on the increment from the residual on the last and second to last iterations. Thus if the residual is denoted with the letter r and an index l representing the sub-iterations, the updated under-relaxation factor is given by

$$\theta^{l+1} = \theta^l \left(- \frac{(r^l - r^{l-1})^T r^{l-1}}{(r^l - r^{l-1})^T (r^l - r^{l-1})} \right) \quad (2.21)$$

In fixed under-relaxation, sometimes too much relaxation can be employed to make the scheme seem more stable, which slows down the computation unnecessarily. This weakness is overcome in the Aitken method since it finds the perfect balance between stability and computational efficiency.

2.4.7 IQN-ILS

With the FSI problem reformulated as a residual function, there can be several methods by which it is solved. If the Jacobian matrix dR/dx was known, Newton iterations could be applied to find a solution [19]

$$\left(\frac{dR}{dx} \right)_{x_n} \Delta x_n = -r^n \quad (2.22)$$

$$x^{n+1} = x^n + \Delta x^n \quad (2.23)$$

This line of thinking leads us to the current state of the art coupling scheme for fluid-structure interaction called Interface Quasi-Newton iterations based on the Inverse Least Squares approximation of the Jacobian (IQN-ILS). The first problem is that the exact Jacobian of the residual is not known, as F and S are not available and a huge system

of equations has to be solved in every iteration [19]. This is already an improvement compared to under-relaxation methods that apply to the entire fluid domain, calculating the complete Jacobian matrix can be computationally expensive. This leads to the approximation part, where instead of calculating the complete Jacobian, we approximate the inverse Jacobian, which is actually all that's required, using a least squares technique. Then, quasi-Newton iterations can be used, for solving the system of equations and obtain a numerical solution [19]

$$x^{n+1} = x^n + \left[\left(\frac{dR}{dx} \right)_{x_n} \right]^{-1} (-r^n) \quad (2.24)$$

and these iterations solve the FSI problem for the fluid-structures interface position. The initial guess for the quasi-Newton iterations is obtained by extrapolating the position of the interface on previous steps [19]

$$x^0 = \frac{5}{2}(x^k) - 2(x^{k-1}) + \frac{1}{2}(x^{k-2}) \quad (2.25)$$

The residual r^n for every inner iteration is calculated as [19]

$$r^n = R(x^n) = S \circ F(x^n) - x^n = \tilde{x}^{n+1} - x^n \quad (2.26)$$

and the solution for this time step has converged when the second norm of this residual is smaller than a convergence criterion ε .

It is apparent from 2.26 that like the complete Jacobian matrix, neither the inverse has to be explicitly calculated. It is possible to simply calculate the product of $\Delta r = 0 - (r^n)$ with this matrix, which can be obtained from previous iterations [19].

The inner residual calculation equation shows that the solution of the two domains in each quasi-Newton equation provides as with two vectors, $\tilde{x}^{n+1}$ and r^n . These two, along with the data from all the previous iterations create two a set of vectors and then each point's difference from the previous step can be calculated as [19]

$$\Delta r^i = r^i - r^n \quad (2.27)$$

$$\Delta \tilde{x}^{i+1} = \tilde{x}^{i+1} - \tilde{x}^{n+1} \quad i = 0, \dots, n-1 \quad (2.28)$$

These two corresponding vectors are stored as the columns of two matrices W^n and V^n along with information from the previous time steps. The vector Δr is approximated as a linear combination of the known Δr^i [19]

$$\Delta r \approx V^n c^n \quad (2.29)$$

with c^n the coefficients of the following decomposition. "The above system of linear equations is an overdetermined system and a least squares solution to this system can then be calculated by using the so called economy size QR decomposition of V^n using Householder transformations" [19]

$$V^n = Q^n R^n \quad (2.30)$$

and from these calculating a coefficient vector c^n by solving the triangular system [19]

$$R^n c^n = Q^{nT} \Delta r \quad (2.31)$$

using back substitution.

In some cases the corresponding equation will have no solution, because the Δr^i vector will be (approximately) linearly dependent in the relevant matrix. In that case, the corresponding c^n is set to zero. Similarly [19]

$$\Delta \tilde{x} \approx W^n c^n \quad (2.32)$$

$$\Delta r = \Delta \tilde{x} - \Delta x \quad (2.33)$$

and by merging these two equations we obtain [19]

$$\Delta x = W^n c^n - \Delta r \quad (2.34)$$

Therefore we have obtained the solution we required [19].

$$\Delta x = \left(\left[\frac{dR}{dx} \right]_{x_n} \right)^{-1} \Delta r = W^n c^n + r^n \quad (2.35)$$

thus getting finally a relation between Δx and Δr by means of the intermediate $\Delta \tilde{x}$ values. This information reuse from previous time steps accelerates the coupling convergence remarkably. The optimal number of previous time steps k that can be reused this way is problem dependent but the change in the rate of convergence near the optimum is insignificant and the method is robust.

2.5 ALE formulation

In fluid mechanics, there are two widely used approaches for describing the fluid particle motion, the Lagrangian and the Eulerian formulation.

Lagrangian algorithms are those in which each computational node of the mesh is associated with one or more particles from the simulated fluid and continuously follows the motion of said particle(s) during runtime. This formulation provides the possibility to actively track free surfaces or moving interfaces during runtime. It is unable, however, to follow large distortions in the computational domain [22].

Eulerian algorithms, on the other hand, employ a computational mesh that is fixed in space and where large distortions in the motion can be handled with ease, at the expense of precise interface and boundary tracking, as well as the resolution of details in the flow [22].

For a wide range of applications, both formulations have their advantages and disadvantages since it is frequent that the situation demands positive characteristics from both formulations to be present in a single computational problem. In this case, working with FSI, there is a structural domain, with a displacement field as a function of the original grid while the fluid has a solution with respect to the current grid, the Euler description. A no-slip condition (similar to a Dirichlet boundary condition) is applied to the deformable domain, which gives the additional condition that the fluid cells near the boundary, that share a face with the solid cells, have the same velocity as the solid cells they share a face with. This implies that the grid points of the fluid move coherently with the points of the solid on the coupling boundary.

There are also static pressure and drag issues in the thin layer surface between the fluid and the solid. To overcome this problem, a technique has been developed known as the Arbitrary Lagrangian - Eulerian (ALE) formulation. When using this method, the

computational mesh is able to move and try to preserve its positive characteristics, such as orthogonality of the cells, while keeping track of the movement of boundaries and free surfaces [23]. One can say that this formulation enables the fluid mesh to be deformed in response to structural deformation. Thus, first apply the traction vector on the solid state and impose the displacement of the fluid boundary on the fluid, re-mesh the grid and interpolate the velocity and pressure on the new grid.

2.6 Motion for the bending vibration of a slender homogeneous beam

The differential equation for the bending of a homogeneous beam with length L is given by equation 2.36 [24].

$$EI \frac{\partial^4 \omega}{\partial x^4} + \rho A \frac{\partial^2 \omega}{\partial t^2} = 0 \quad (2.36)$$

where ω stands for the eigen-frequencies, ρ is the mass density, A is the area of the beam and EI is a quantitative measure of the material's stiffness, comprising of the inertial moment and young's modulus, which is a quantitative measure of the elasticity of a material.

With separation of variables, assuming that the solution is of the form $\omega(x,t) = X(x)T(t)$ equation 2.36 can be written as

$$\frac{X^{(4)}}{X} = -\frac{\rho A}{EI} \frac{T''}{T} = \lambda \quad (2.37)$$

which is an eigenvalue problem with λ the eigenvalues. This formulation leads to

$$X^{(4)} - \mu^4 X = 0, \quad \mu^4 = \lambda \quad (2.38)$$

Solving this gives solutions of the form:

$$X_n = c_n((\sin \mu_n L + \sinh \mu_n L)(\cos \mu_n x - \cosh \mu_n x) - (\cos \mu_n L + \cosh \mu_n L)(\sin \mu_n x - \sinh \mu_n x)) \quad (2.39)$$

For each eigenvalue λ there is a differential equation for T from 2.37

$$T'' + \omega_n^2 T = 0, \quad \omega_n^2 = \lambda_n \frac{EI}{\rho A} \quad (2.40)$$

The general solution is of the form

$$T_n = \alpha_n \cos \omega_n t + \beta_n \sin \omega_n t \quad (2.41)$$

Using 2.39 and 2.41 we can get the solution to 2.36 as

$$\omega(x,t) = \sum X_n T_n \quad (2.42)$$

These eigen-frequencies are also called the natural frequencies of the beam. Additionally, the natural or normal modes of the beam can be calculated by finding the roots of equation 2.39 and the natural frequencies correspond to the solution of the eigen-frequency equation for $X_n = 0$.

3. Implementation

3.1 Meshing

The first and most important thing when it comes to CFD applications is having a correct and well-structured mesh. For this study, the *OpenFOAM* tool *snappyHexMesh* has been used for the mesh generation [8]. "*SnappyHexMesh* is an utility that generates 3-dimensional meshes containing hexahedra and split-hexahedra automatically from triangulated surface geometries in Stereolithography (STL) format" [25]. The mesh is generated in an iterative way, creating a rough block mesh, bigger and coarser than the one we want to achieve. Then, using repeated morphing iterations, the mesh slowly conforms to the surface. There are additional optional phases for adding cell layers near the wall boundaries and the final quality of the mesh can be defined precicely by the user. It also has support for parallel execution [25]. The STL geometry files where exported from the open-source software *Salome*, which was used to create the original CAD model of the computational domain.

In order to run *SnappyHexMesh* the following are required [25]:

- Surface data files in STL format.
- A background hex mesh, which defines the extend of the computational domain and a base level mesh density. This can be easily generated using another OpenFOAM utility called *blockMesh*.
- A *snappyHexMeshDict* with appropriate entries which will define the properties of the created mesh.

The utility automatically follows the instructions specified in the dictionary file and handles the mesh generation, starting from the initial background hex mesh and repeatedly splitting the cells in the locality of the provided surfaces. Then, a cell removal stage follows, where the cells in the regions that are not needed are removed (either the inner or the outer region in relation to the surfaces) depending on the properties defined. Afterwards, a further refinement step follows, where the remaining cells (in the region of interest) are further split in relation to one of the surfaces or an externally provided volume or surface. The mode of refinement is modifiable and the refinement step can happen either inside, outside or at a specified distance from the volume/surface.

The next step is more sophisticated, and includes a surface "*snapping*" process. This step handles cells that "cut" into the surfaces by removing a part of them or moving the mesh in such a way that does not alter its edge/vertex features.

Finally, the last step is optional, and it provides the possibility to add additional *layers* of hexahedral cells aligned to the boundary surfaces by shrinking the existing mesh from the boundary and inserting layers of cells. This is a useful feature to make sure that we have the proper "wall treatment" for the viscous sub-layers near the flow boundaries.

As seen in figure 3.1, multiple cell layers have been added in the viscous sub-layer near both the outer walls and the tube, to guarantee a correct solution due to viscosity near the boundaries. A further cell splitting step has been performed in a cylindrical area around the housing tube in regards to the rest of the computational mesh, for better results.

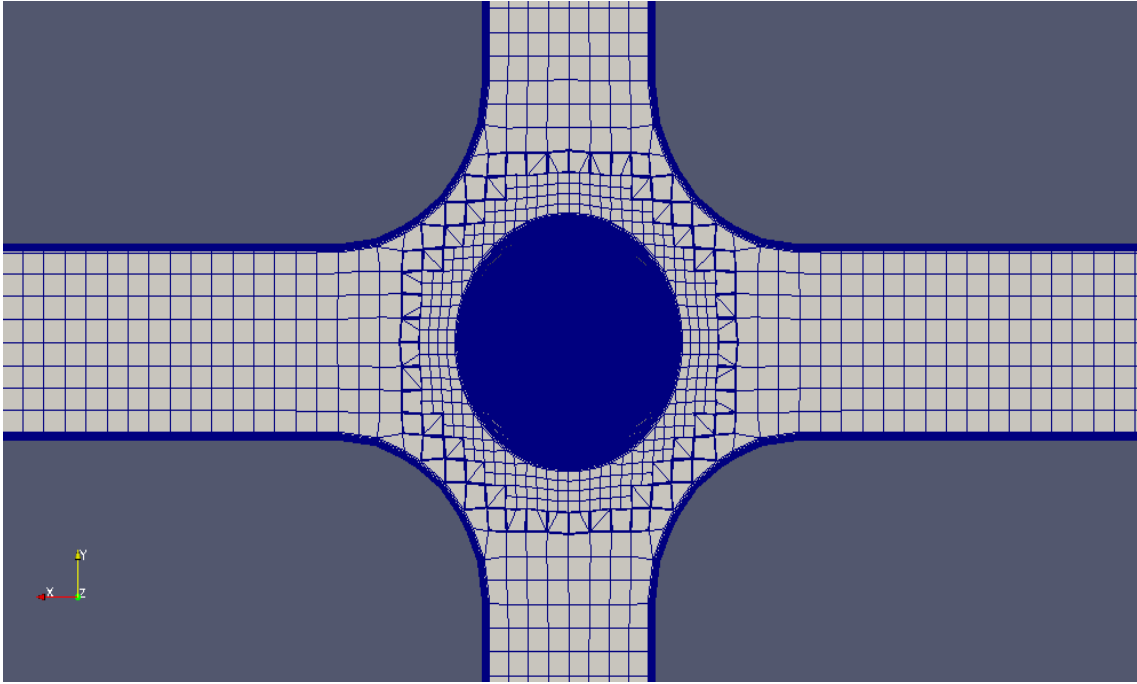


Figure 3.1: Computational mesh seen from a plane cut in the flow normal direction.

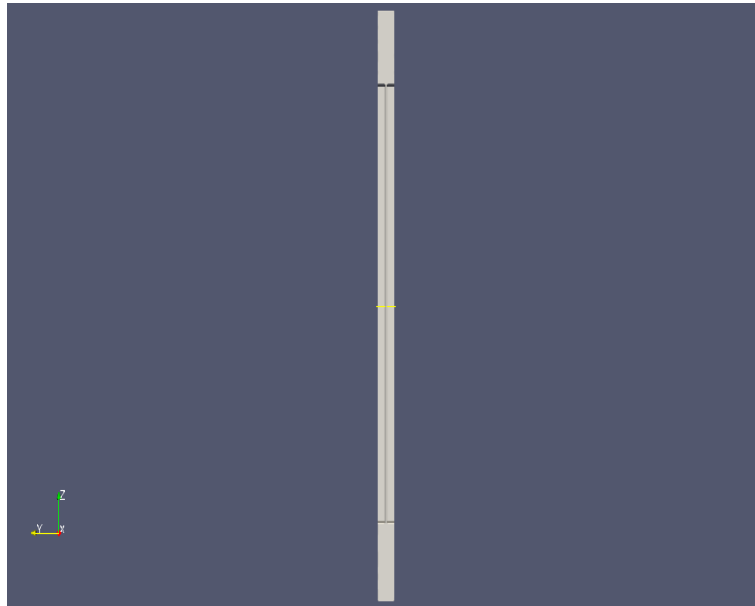


Figure 3.2: Computational mesh as seen from the outside.

3.2 Turbulence

The first thing that requires an evaluation is how the geometry is going to behave against a developed turbulent flow, without considering any vibrations or added deformations caused by the structural response. This will allow examination of the velocity and pressure fields, assumptions about the cause of the behavior of the structural mesh and comparison of the results to the experimental setup and a validation of the flow field. Finally, it will provide data for the initial and boundary conditions of the various numerical fields in a

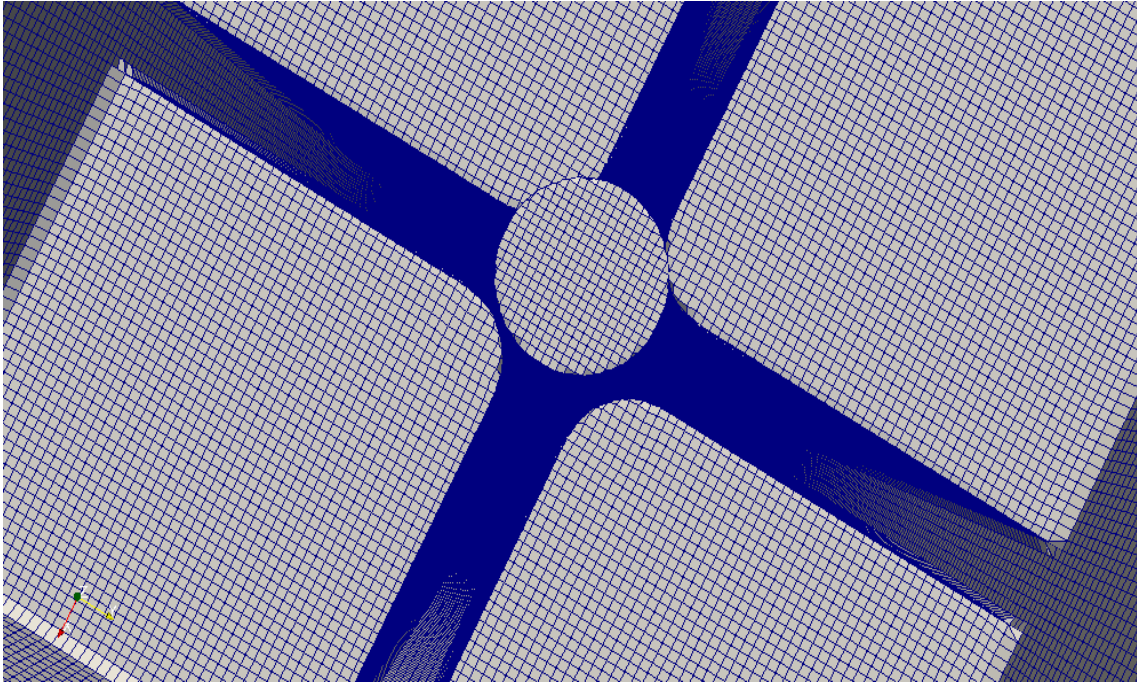


Figure 3.3: Computational mesh as seen from the inside.

developed flow. These data can be re-used, by means of a "mapping" technique, on the second part of the study, that will include the FSI, as a way to accelerate the process.

There are extensive methods to approach turbulence when working with *OpenFOAM*. *OpenFOAM* has implemented available solvers for a high variety of situations, including RANS, LES, DES (Detached Eddy Simulations) and even Direct Numerical Simulations (DNS). This study is going to focus solely on high accuracy models (DNS and LES). DNS is not computationally feasible (at least in the time frame of this project), since this study examines a high turbulence flow field with Reynolds number $Re > 10000$ and RANS models are not physically acceptable and are mostly used for engineering applications where the interest lies in the calculation of specific values and not on the flow field itself. Thus, the best course of action is to use an LES model. *OpenFOAM* offers a high variety of LES models, including the common and oldest Smagorinsky - Lilly SGS model. For this study, the *oneEqEddy* model is used, which is an implementation of an eddy-viscosity model for the k-equation [16]. In one-equation models like this, a Partial Differential Equation (PDE) is derived for the turbulent kinetic energy and the unknowns (turbulent viscosity and conduction coefficient) are expressed as a function of the turbulent kinetic energy, along with models for any additional unknown correlations. The difference between the one equation model and the Smagorinsky - Lilly model lies in the fact that the one equation model is based solely on how the turbulent kinetic energy is modeled, which comes with an equation for the transport of the turbulent kinetic energy implemented.

3.2.1 Turbulence with OpenFOAM

Setting up a simulation with *OpenFOAM* is pretty straightforward [26]. The software is terminal-based with the necessary files organized in entry files in a case directory. There are 3 common directories that are required to setup most (if not all) common *OpenFOAM* simulations. First, a *0* directory is required, that includes entry-files for all scalar and

vector fields necessary for the computations containing boundary conditions for all the boundary surfaces and initial conditions for the inner domain. Next, a *constant* directory is required that includes necessary files with properties that remain constant throughout the simulation, like transport properties, the computational mesh etc. Finally, a *system* folder should be present that includes files with information about the schemes and solutions used in the simulation, as well as dictionaries for additional utilities that indirectly influence the case and a *controlDict* dictionary file that includes all the basic user defined controls needed, like the time step, the output interval, the data write format etc. Almost all solvers in *OpenFOAM* are based on the Finite Volume (FV) method. The solver used in this study for turbulence is the *pisoFoam* solver. *PisoFoam* is a transient solver for incompressible flow, including generic turbulence modelling and using the PISO algorithm (Pressure Implicit with Splitting of Operator) for solving the Navier-Stokes equations [8].

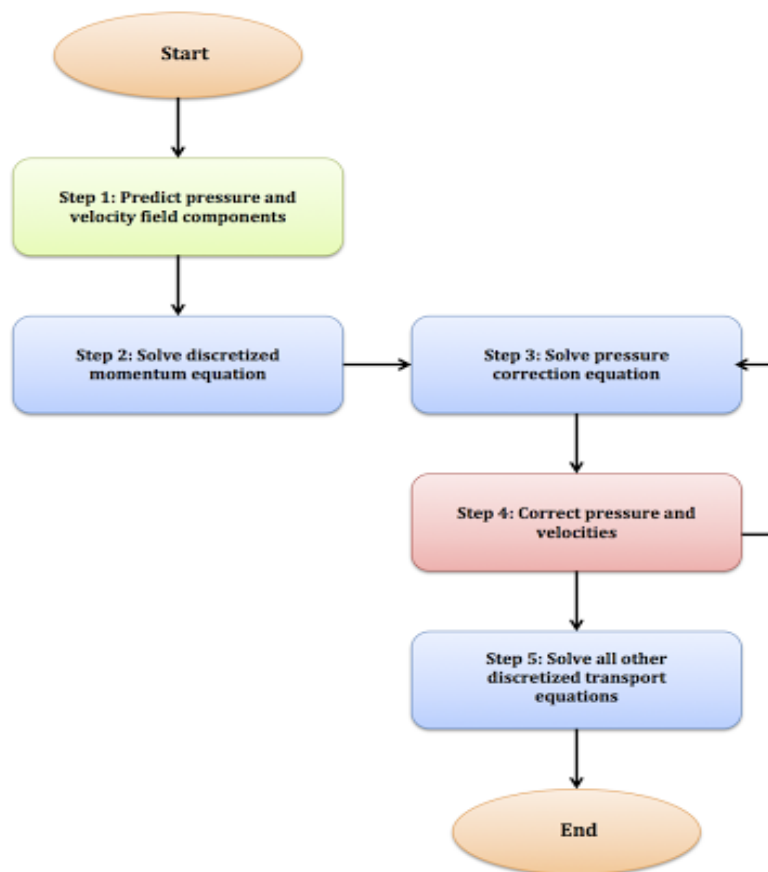


Figure 3.4: PISO algorithm flow chart.

This algorithm was proposed by Issa in 1986 without iterations and with large time steps and less computing effort [27]. It is designed to work as seen on the flowchart of figure 3.4. It is a pressure-velocity calculation procedure for the Navier-Stokes equations.

PISO involves one predictor step (step 1 - see figure) and two corrector steps for 2nd order accuracy (step 4 - see figure) and is designed to satisfy mass conservation using predictor-corrector steps.

Next, we have to consider the boundary conditions. We need to define a boundary condition for every boundary surface of the geometry that we want to differentiate in some way from the rest. Thus, we need separate boundary conditions for the inlet, outlet and one the rest of the walls. For this case, the boundary conditions have been set as follows.

For the velocity:

- First, a constant velocity flow rate has been set at the bottom surface of the geometry, named inlet. The experimental setup had provided measurements for 5, 10 and 15 m^3/s . For validation purposes and simplification only the case of 10 m^3/s has been simulated. This boundary condition is called *flowRateInletVelocity* in *OpenFOAM*.
- An *inlet/outlet* boundary condition has been defined for the velocity of the top surface of the geometry, named outlet, which specifies free flow at this surface, allowing negative flow rate (and thus backtrack of flow through this surface).
- The rest of the boundaries (walls) have been fixed at a constant velocity of zero.

For the pressure and turbulent kinetic energy:

- A *zeroGradient* boundary condition has been specified for the pressure for all boundary surfaces. This is the equivalent of a Neumann boundary condition in *OpenFOAM*. An exception is the outlet boundary surface that has been fixed at a zero total pressure, to act as a reference for the total pressure in the computational domain.
- Additionally, a fixed value of 0.4 m^2/s^2 has been set for the kinetic turbulent energy at the inlet. This is a reference approximate value, that has been roughly calculated from the mechanical properties of the geometry to help the kinetic energy stabilize to its final value faster and has no physical meaning.
- All other fixed values are set to zero.

All the boundary condition information have been summarized in table 3.1. A kinematic viscosity value of $10^{-5} m^2/s$ has been specified for the whole computational domain as a fluid property in the *constant* directory as well as a variable that specifies that the transport model is Newtonian and the necessary constant variables for the turbulence model. Additionally, the computational schemes have been assigned in the *fvSchemes* file in the *system* directory to appropriate schemes. A backward first order scheme has been chosen for the d/dt discretization and a Gauss general linear method for the divergence, gradient and laplacian discretizations. Finally, a sufficiently small time step (10^{-5}) has been defined in the control dictionary, in order for the *Courant-Friedrichs-Lewy* (CFL) condition to be satisfied throughout the computational domain during the simulation. This is a necessary mathematical condition in order for convergence to be guaranteed when solving certain partial differential equations (PDE).

The solver has been set to run and collect data at a fixed interval of 1 second, for 3 separate cases with varying mesh size. The first one had a low refinement rate with a computational mesh of approximately 1,5 million cells, the second case had a higher refinement rate with 7 million cells and the last one had an extremely accurate mesh of 20 million cells. A spectral analysis has been performed on all 3 separate cases, in order to examine how the size of the computational mesh affects the results.

Quantity	Inlet	Outlet	Walls
U	flowRateInletVelocity	inletOutlet	fixedValue
p	zeroGradient	totalPressure	zeroGradient
k	fixedValue	inletOutlet	fixedValue
nuSgs	zeroGradient	zeroGradient	zeroGradient

Table 3.1: *Boundary condition specification for the turbulence study.*

3.3 Sampling

The next thing that requires a consideration is the strategy used to sample data during the simulations. The strategy used needs to be adjusted to match the experimental data collected in order to have the ability to perform a validation of the simulation against the experiment afterwards. Thus, knowledge of the existing experimental sampling strategy is required.

3.3.1 Experimental setup sampling

The main numerical fields of interest are velocity and pressure in this kind of simulations. For pressure measurements, pressure transducers have been placed in the narrow flow path between the tube and the plates forming the corners of the fuel bundles. The differential pressure transducers have a range of 100 *mbar*, an uncertainty of $\pm 0.5\%$ FSO (Full Scale Output) (corresponding to 0.5 *mbar*) and a response time of < 5 *ms*. The transducers are coupled to a common pressure tap with a valve, at the level of the transient pressure measurements. The valve is kept open until the flow rate and static pressure is stabilized and then closed while the measurements are made. This way, it is made possible to use sensitive differential pressure transducers like this although the ambient pressure at some points was way too high for their physical tolerance. The sampling rate was at a rate of 1 kHz. Velocity on the other hand was measured using electromagnetic mass flow measurements. The water flow rate is between 0 and 15 *l/s* which corresponds to a bulk velocity of approximately 0-15 *m/s* in the testing section. Multiple measurement locations have been chosen throughout the geometry for pressure, at the same position described but at different heights throughout the flow channel. The height positions chosen were at 100, 500, 900 and 1300 *mm* from the beginning of the flow channel, as shown in figure 3.5 [1].

3.3.2 Sampling with OpenFOAM

Sampling in *OpenFOAM* can be done in various ways [8]. Important vector and scalar fields that are calculated during run-time and are required on the next time step, are written out at regular time intervals during the simulation, specified by the user in the control properties. This category includes fields like velocity, pressure, turbulent kinetic energy and sub-grid scale viscosity, for a turbulence simulation like this case. However, fields that are not calculated by default during run-time have to be explicitly specified to be calculated and written out. *OpenFOAM* provides the possibility for a large scale of optional field calculations. Some of them can be calculated after the end of the simulation, from other fields (already calculated during runtime) with utility functions created for this pur-

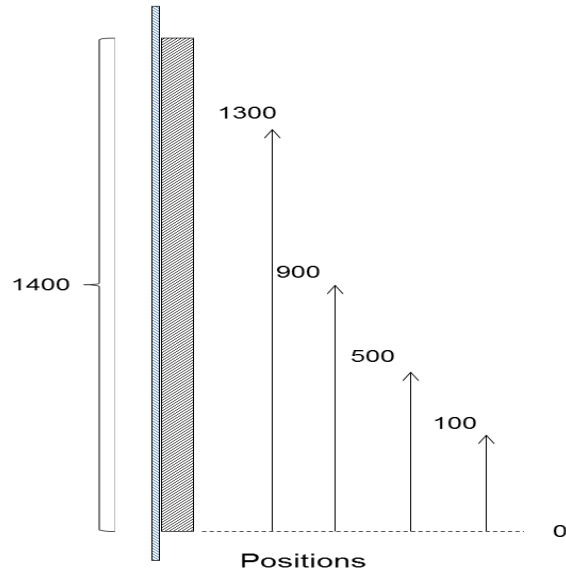


Figure 3.5: Location of pressure transducers for pressure measurements in experimental setup.

pose. For example, the field u' can be approximately calculated from the turbulent kinetic energy as $\sqrt{2k/3}$. Others can be marked for calculation by the user with external function objects, located in the control dictionary file. Possible sampling methods include:

- Surface sampling, providing different field values on the cells adjacent to boundary surfaces
- Probe sampling, where probes are placed at specified locations throughout the computational domain and fields of choice are calculated and written out in data files for specified intervals
- Field averaging, where mean and fluctuating values are calculated for specified fields.

Our interest lies in the last two methods. Probes can be set in various locations to calculate the velocity and pressure there, as well as at the corner of the fuel bundles where the experimental measurements took place, so that proper comparison can take place between the two cases. Also root mean squared (RMS) and prime values for the velocity and pressure are of interest, thus a field averaging function will be specified as well for these two fields.

3.3.3 Field Averages

Field averages are of interest along lines, contours or volumes within the computational domain rather than on specific points. For easier analysis, specific lines at multiple heights throughout the flow channel have been chosen for calculation and plotting of the velocity and pressure average values. These lines can be seen on the figure 3.6 and measurements have been taken at the same heights as the transient pressure measurements of the experimental setup.

3.3.4 Probe sampling

Probe sampling is important for obtaining time data series, specifically velocity and pressure, at specific points, which are necessary for spectral analysis of the results. Having measurements at a constant sampling rate throughout the duration of the simulation, the

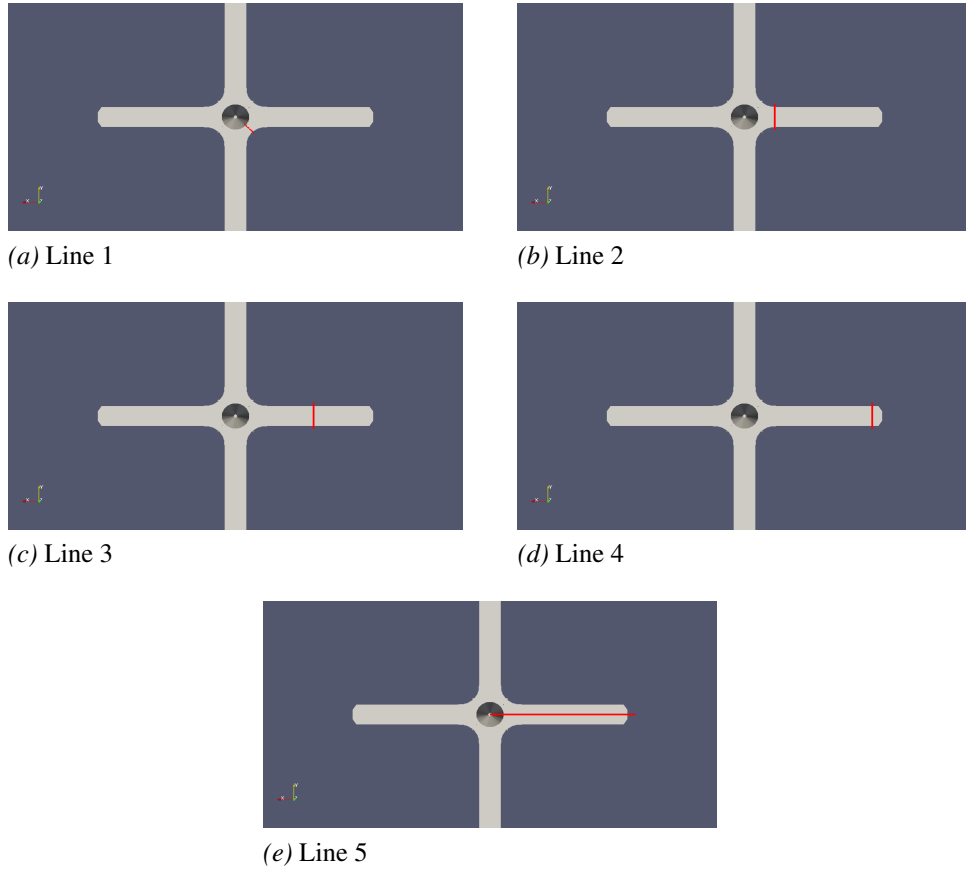


Figure 3.6: Line locations for RMS and variance plotting.

possibility to transform the data to the frequency domain, by use of Fourier transforms, arises and obtain Power Spectral Density (PSD) plots that allow examination of vibration frequencies inside the flow channel and give further insight about the circumstances under which those FIV occur. Probe locations are shown with red dots in figure 3.7 for a specific height and have been placed in the same manner with other monitoring locations at multiple heights, coinciding with the heights at which the transient pressure measurements have been taken for the experimental setup. Of course, the way the data are sampled for the velocity that is a vector field, it is possible to only analyze the spectrum of one of the three vector components at a time, so separate frequency spectrums for each direction. Pressure, however, is a scalar field defining the whole control volume in all directions, making it able to visualize the frequency spectrum for it in one representation. The sampling happens at every time step t_s thus there is a sampling rate (frequency) of $f_s = 1/t_s$, which represents half of the maximum possible frequency value our analysis is sensitive to (considering the symmetry of the Fourier space). A high enough sampling frequency (low time step) is required in order to be able to get an acceptable frequency range. In these cases, a time step in the order of 10^{-5} has been taken thus a sampling frequency in the range of 10 to 100 kHz is obtained.

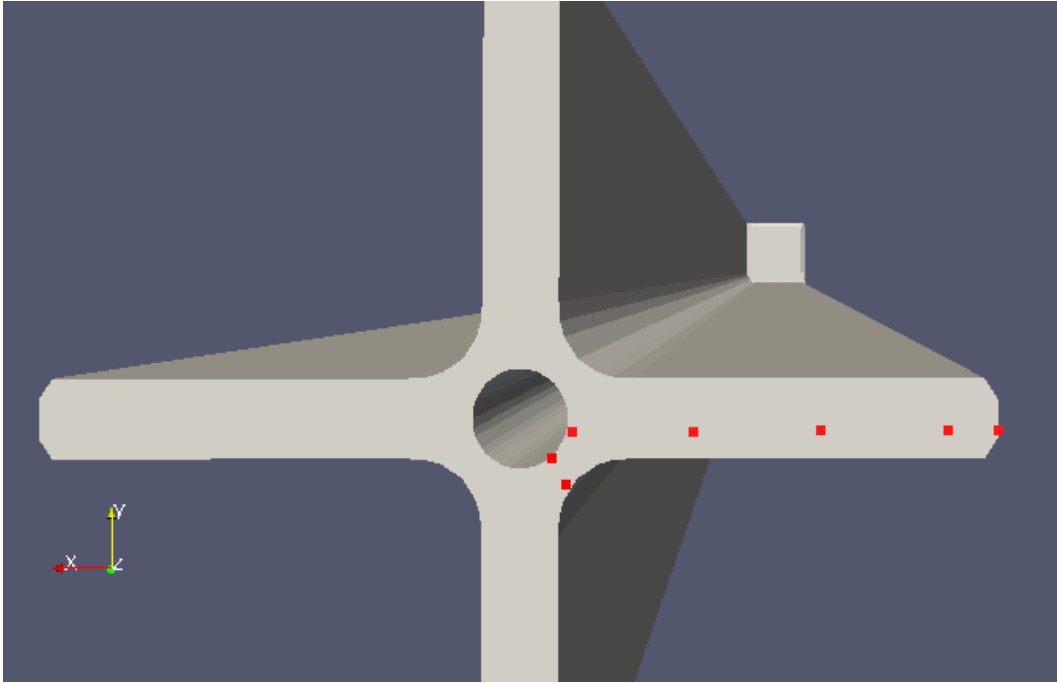


Figure 3.7: Locations of probes inside the computational domain for sampling of time-series.

3.4 Fluid Structure Interaction

The next step involves creating a structural mesh for the deformable domain of the geometry, namely the housing tube in the center of the computational domain. This will allow the activation of the FSI on the interface between the two meshes.

3.4.1 FSI with OpenFOAM

Fluid-Structure Interaction in *foam-extend* is implemented with a partitioned approach, in the form of black-box fluid and structural solvers that are coupled on the fluid-structure interface. Two separate solvers are used, one for each domain and then a General Grid Interface (GGI) interpolation technique is used between the two solvers to map the shared interface to each other and handle the dynamic mesh motion [18]. GGI is used to couple interfaces in the computational domain where the nodes on each side of the interface do not match exactly for every moment during the simulation. This is a result of the mesh motion and weight factors are used to decide how much information should be transferred from one side of the interface to the other.

Two FSI solvers included in the latest *foam-extend-3.1* version have been tested, one of them being the older *icoFsiNonElasticLinULSolidFoam* that came with the release of the software and the other one of them being offered externally in an additional package called *extend-bazaar* and named *fsiFoam*. The second solver was chosen in the end for obtaining the final results of this study. It is more advanced, being the predecessor of the older one and had higher functionality, including pre-implemented turbulence modeling for the fluid domain, in the form of a solver based on the *pisoFoam* solver of the standard *OpenFoam* release. A general algorithm for *fsiFoam* is underlined in figure 3.8.

The first course of action is generating the structural mesh required for the FSI simulation. This can be achieved in OpenFOAM by use of an utility tool called *extrudeMesh*. What this tool does is to extrude the mesh in a specified fashion (linearly in the surface

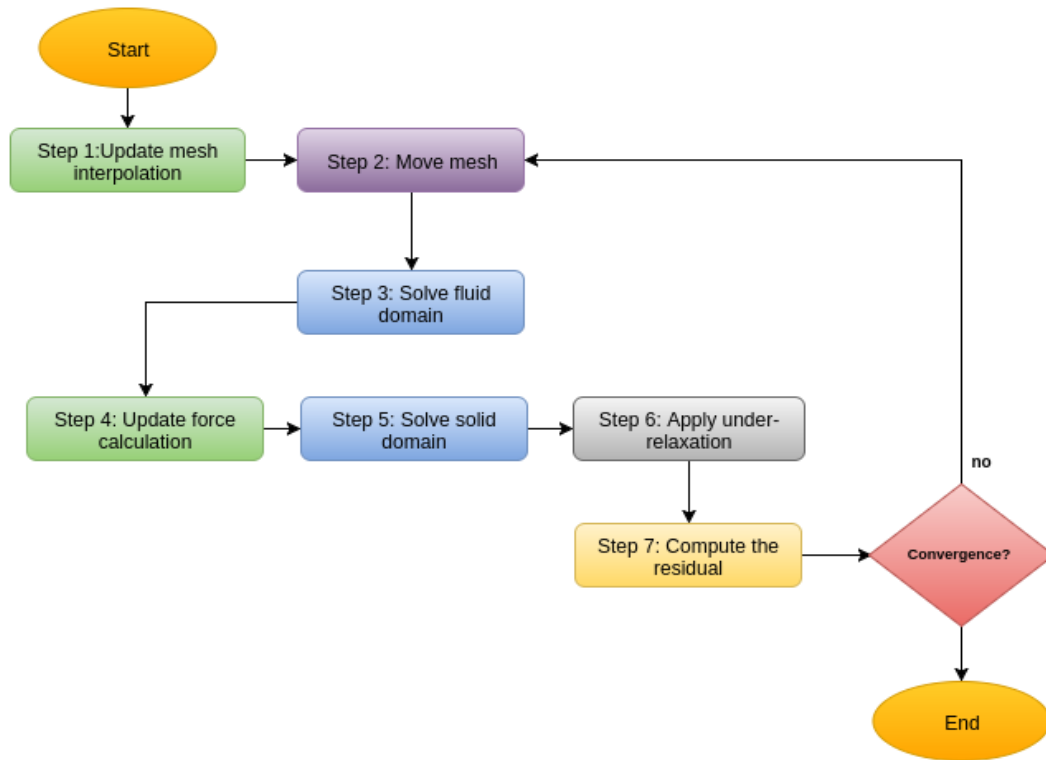


Figure 3.8: *FsiFoam* algorithm flow chart.

normal direction, linearly in a specified direction, radially etc) from a specified surface patch. By using this tool on the existing fluid mesh and extruding linearly inwards for a specified length for the tube patch, a cylindrical mesh can be created that conforms perfectly on the existing fluid mesh and thus on the fluid-structure interface and can be used as a structural mesh for the FSI simulation. Having a perfectly conformal mesh is a really nice property, since we are dealing with an unstructured mesh and it is normally impossible to create a perfectly conformal structural mesh in such a case. The extrusion was specified to happen for 0.6mm since that's the actual thickness of the test tube in the experimental setup, with a minimum of three layers of cells created to ensure correct motion in all three dimensions.

Having created the structural mesh, the next step requires to set the case folder for the simulation. Two separate folders are required inside the original case folder, named *fluid* and *solid* for the two meshes respectively, as the name suggests. Both folders require all the files and folders a normal simulation would need, like two separate cases. Additionally, a few extra property files are required in the fluid domain, that contain the necessary information the mesh motion algorithm and FSI need. Additionally, the faces that belong to the fluid-structure interface need to be explicitly specified for both meshes, so that the GGI utility knows which cell faces to try to interpolate and to where. This is possible with the use of an additional utility tool, called *setSet*, which allows the massive selection, manipulation and handling of surfaces, cells and points within the computational domain.

For the mesh motion, a dynamic mesh motion solver for a finite volume mesh has been used. The solver is called *velocityLaplacian* and the way it works is to calculate the displacement of the points in the computational mesh with velocity specified as the initial and boundary condition of the laplacian equation. The diffusivity model is specified to

Quantity	Fluid Mesh			
	Inlet	Outlet	Tube	Walls
U	flowRateInletVelocity	inletOutlet	movingWallBC	fixedValue
p	zeroGradient	totalPressure	extrapolatedPressure	zeroGradient
k	fixedValue	inletOutlet	fixedValue	fixedValue
nuSgs	zeroGradient	zeroGradient	zeroGradient	zeroGradient

Table 3.2: *Fluid mesh boundary conditions for FSI simulation.*

calculate the "*stiffness*" of the mesh to the motion solver based on the inverse distance from the fluid-structure interface. This means that the further you get from the interface, the weaker the coupling becomes, in order to be able to provide a physically correct motion. Both Aitkens under-relaxation and IQN-ILS coupling have been tested, with the latter providing better results as convergence is achieved in fewer outer correction steps.

Finally, additional properties had to be modified compared to the turbulent case due to the FSI both for optimization and correctness. Specifically, the boundary conditions had to be modified. The boundary patch corresponding to the housing tube have to be separately explained from the rest of the wall boundaries, and split in 3 different patches.

For the solid domain:

- The bottom and top edges of the cylindrical structure had to be cut off the rest of it and the correct boundary conditions had to be defined. The bottom patch, named *tube-inlet* had to be fixed in place, using a *fixedDisplacement* velocity value of zero.
- The top patch, named *tube-outlet*, on the other hand, had to be "pinned", a boundary condition which approximately has the same behavior as if a spring kept it attached to the core grid. This is achieved with an implemented in *foam-extend* boundary condition called *fixedNormalDisplacement*, and set initially to zero as well. This will allow vibration on the face normal direction of the top patch but confined by the flow field.
- Finally the rest of the tube had to be set as *tractionDisplacement* which calculates the displacement of the cells based on the traction force due to the pressure and velocity of the flow field on the cells adjacent to the tube.

For the flow domain:

- The pressure field for the patches corresponding to the tube had to be changed from zeroGradient which corresponds to an inner extrapolation of the fluid field to *extrapolatedPressure* which corresponds to a boundary condition that performs both an inner extrapolation of the flow field and an outer calculation depending on the values of the solid field calculation, depending on the step of the calculation.
- Additionally, the velocity field for the free part of the tube had to be switched to *movingWallBC* which corresponds to the boundary condition for a patch that is allowed to move freely based on the traction force on it.

The boundary conditions explained in detailed above are summarized in tables 3.2 and 3.3.

3.4.2 Fundamental frequencies of structural domain

The fundamental frequency of a beam with uniformly distributed mass, f_{s0} , [28] [24] is given by equation 3.1

Solid Mesh			
Quantity	Tube-inlet	Tube-outlet	Tube-moving
DD	fixedDisplacement	fixedNormalDisplacement	tractionDisplacementIncrement
pointDD	fixedDisplacement	calculated	calculated

Table 3.3: Solid mesh boundary conditions for FSI simulation.

$$f_{s0} = \frac{1}{2\pi} \frac{K_n}{L^2} \sqrt{\frac{EI}{\rho_s}} \quad (3.1)$$

where E is Young's modulus, L is the tube length, I is the area moment of inertia and ρ_s is the mass density (mass/length). K_n is a constant depending on the boundary conditions of the beam. They can be calculated by solving the eigenvalue problem for the differential equation for the bending of a slender beam. For "fixed-free" and "fixed-pinned" boundary conditions this constant is $K_1 = 3.5156$ and $K_1 = 15.418$ respectively. These boundary conditions are those used for the simulations, meaning that the bottom end of the beam will always be *fixed* on the grid in its initial position while the top end will either be "free", meaning it is free to move in any direction, or "pinned", meaning it is restricted in the x and y directions but it is allowed to vibrate in the z direction, as if supported by a spring, which is the actual physical behavior that this study wants to represent. The experiment itself involved a *pinned* top boundary condition, which will be the main interest of this study. The area moment of inertia for a thin wall tube is $I = \pi/4(r_{outer}^4 - r_{inner}^4)$ with $r_{outer/inner}$ the outer and inner radius respectively. Using a stainless steel tube with 8mm diameter and 0.6mm wall thickness, $E = 197GPa$, $\rho = 7863kg/m^3$, $\rho_s = 0.11kg/m$ and $L = 1.486m$, a fundamental frequency for the "fixed-pinned" boundary conditions of 14.6Hz for air flow and 10.1Hz for water flow is obtained. This is calculated experimentally by measurements when the tube is bent and then released and left to vibrate freely in its natural frequencies. Then the rest of the natural frequencies of the tube can be calculated by using the K_n ratio and the correlation equation between the natural modes of the beam

$$f_s = f_{s0} \frac{K_n}{K_1} \quad (3.2)$$

Using the analytical values for the K_n constants, the natural frequencies of the tube are summarized in table 3.4 for the first 6 natural modes in both air and water flow. Note that even though the data for air flow are provided and there exist experimental data for such a flow, in the FSI simulation only water flow have been taken into account.

mode	K_n	K_n ratio	frequency(air)(Hz)	frequency(water)(Hz)
1st	15.4	-	14.6	10.1
2nd	50	3.246753	47.4026	32.79221
3rd	104	2.08	98.5974	68.20779
4th	178	1.711538	168.7532	116.7403
5th	272	1.52809	257.8701	178.3896
6th	385	1.415441	365	252.5

Table 3.4: *Table of the natural frequencies of the housing tube in air and water flow.*

4. Results & Discussion

In this section, the results obtained from this study are presented and explained. For the turbulence study, three computational meshes of different resolution have been tested, one with 1.5 million cells, a second one with (approximately) 8 million cells and an extremely fine mesh with 20 million cells. Time data series have been obtained from probe sampling as explained in chapter 3. These data are going to be used to obtain frequency spectra of the velocity and pressure fields and compare with the analytical expressions of the tubes' natural frequencies [29], in order to find some correlation between them that will suggest the existence of sub-critical vibrations, flow-periodicity resonance between the turbulent vibration frequencies and the natural frequencies of the tube or fluid instabilities, leading to FIV [30] [3] [7]. For the FSI simulation, only one computational mesh with approximately 4.2 million cells for the fluid and 250 thousand cells for the solid has been examined. Other computational data and validation methods are going to be presented and explained. Finally, the HPC capabilities of the software is going to be examined.

4.1 Turbulence

A representation of the magnitude of the velocity field on a vertical cut in the direction of the flow inside the computational domain can be seen on figure 4.1. The turbulence is clearly visible near the outlet, where the flow had developed after passing through the narrow flow channel between the housing tube and the fuel bundles. The edges of this narrow flow channel are clearly visible as well where the switch from low to high velocity and reversed is located. Furthermore velocity flow harmonic perturbations can be observed around the tube's surface, which hint the possibility of turbulence buffeting as a cause for the FIV.

In figure 4.2 a plot of the mean, variance and instantaneous velocity at the time of the ending of the simulation (after 1 second of real time) across a line parallel to the z axis and in the middle between two fuel bundles and between the tube and the boundary surface can be seen. The turbulence is clearly observed by the oscillation of the magnitude of the velocity about the mean velocity value.

4.1.1 RMS values and variances

For this part, data were plotted for velocity and pressure across specific lines, as explained in chapter 3. In figures 4.3 the velocity magnitude can be seen, along with its mean value and variance profile across all 5 different lines for the same height along the housing tube, at 900 mm from its base. Since there is a bounded channel flow, its effects can be clearly seen in the plots [15]. For the first few millimeters, the *law of the wall* holds, as formulated by Prandtl in 1925. According to this law, very close to the wall, there is an **inner layer** in which the mean velocity profile is determined only by the viscous scales, independent of the channel width or the bulk velocity of the channel flow. Thus, the dimensionless velocity:

$$u^+ = \frac{\langle U \rangle}{u_t} \quad (4.1)$$

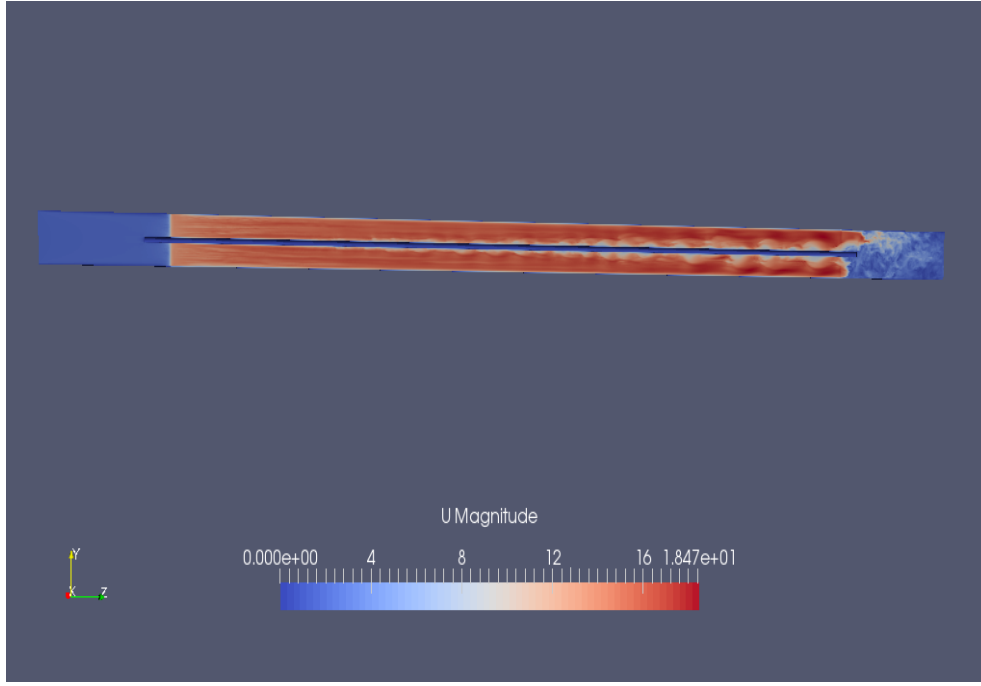


Figure 4.1: Magnitude of the velocity inside the computational domain after 1 second of simulation.

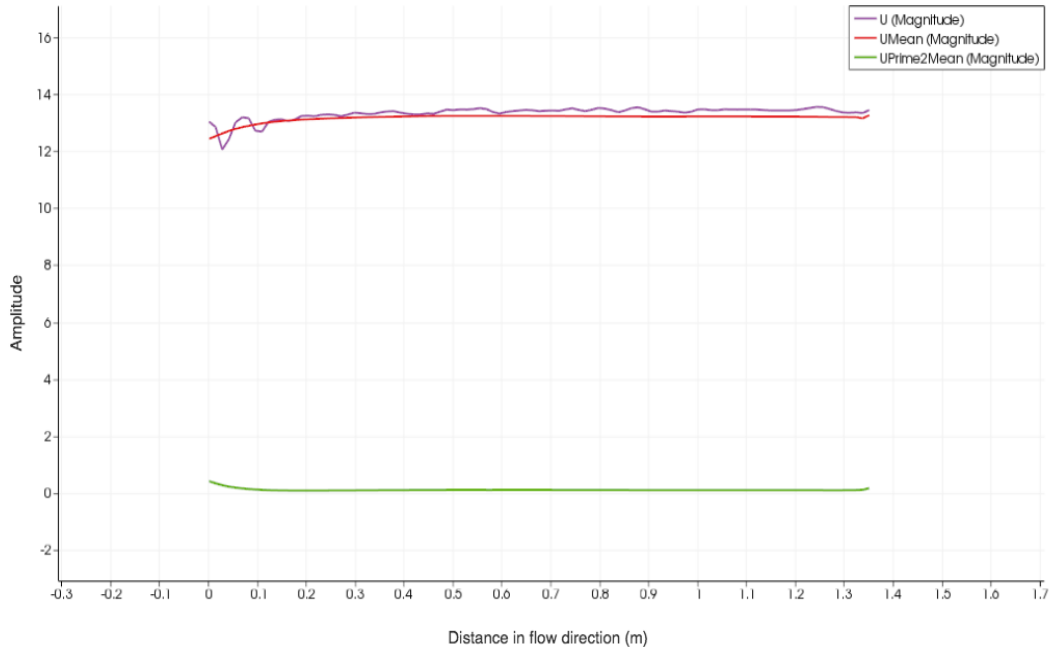


Figure 4.2: Velocity profile in flow direction after 1 second of simulation

where $\langle U \rangle$ is the mean velocity and u_t is the friction velocity near the wall, is dependent only on the distance from the wall, measured in viscous lengths (or wall units) and denoted by:

$$y^+ = \frac{u_t y}{\nu} \quad (4.2)$$

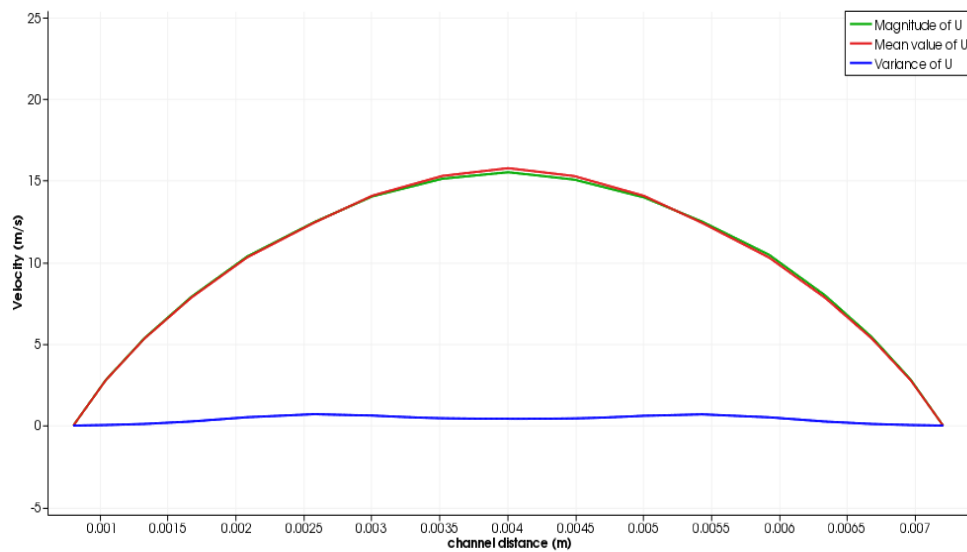
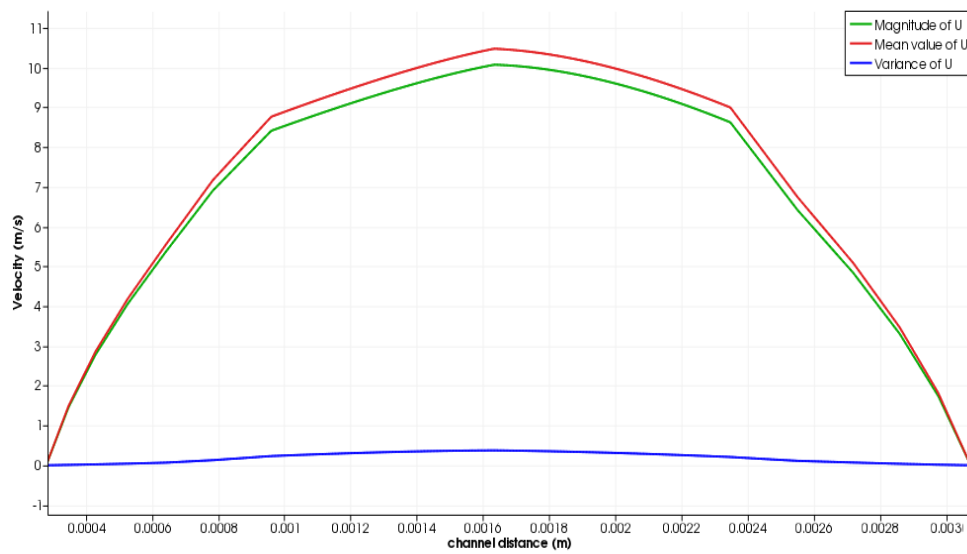
where y is the channel width and ν the kinematic viscosity. For high Reynolds numbers, the outer part of the inner layer corresponds to high y^+ values, where it can be supposed

that viscosity has little effect and the velocity profile develops a logarithmic relation with the distance from the wall, referred to as the *logarithmic law of the wall*, or simply the *log law* and the region is referred to as the **outer layer**. In mathematical form, this can be expressed as:

$$u^+ = \frac{1}{k} \ln y^+ + B \quad (4.3)$$

where B is a constant and k is the von Karman constant.

In the region between these two laws, there exists a smaller layer called the **buffer layer**, which is a transition region between the viscosity and turbulent-dominated parts of the flow. These three regions can be clearly distinguished in the plots on both sides due to the closed channel flow.



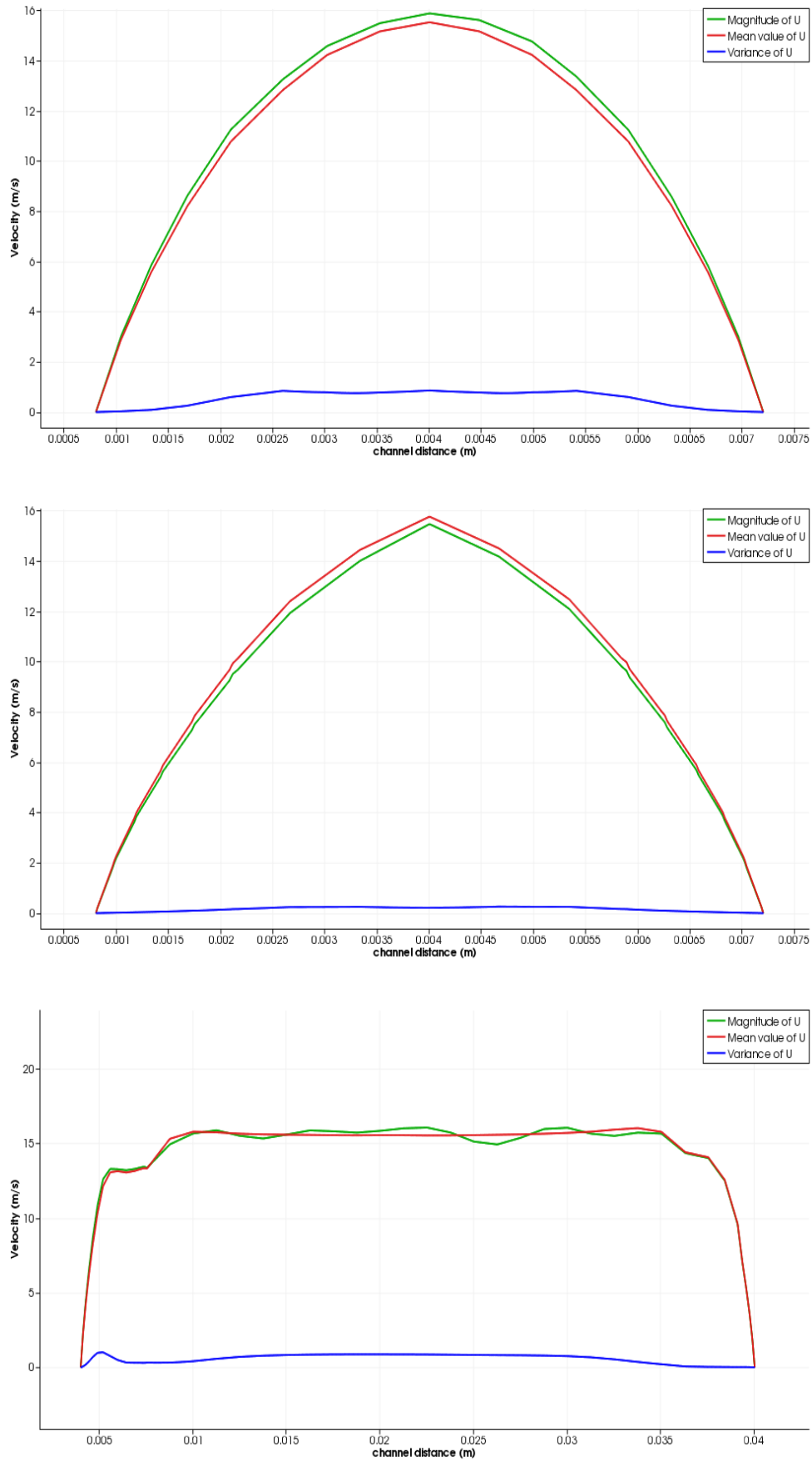
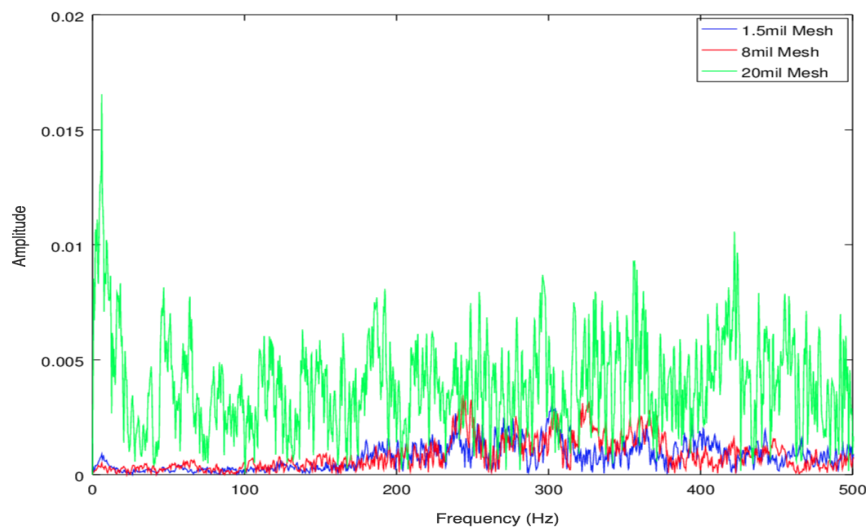


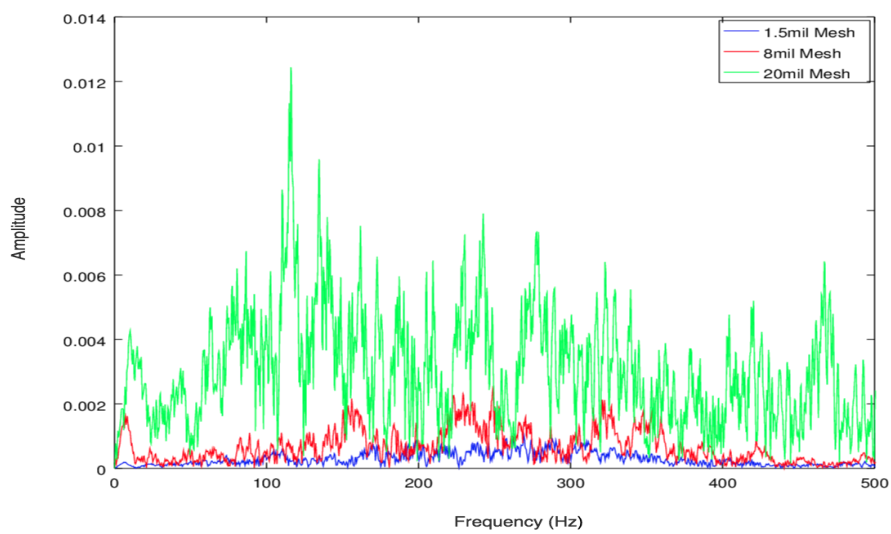
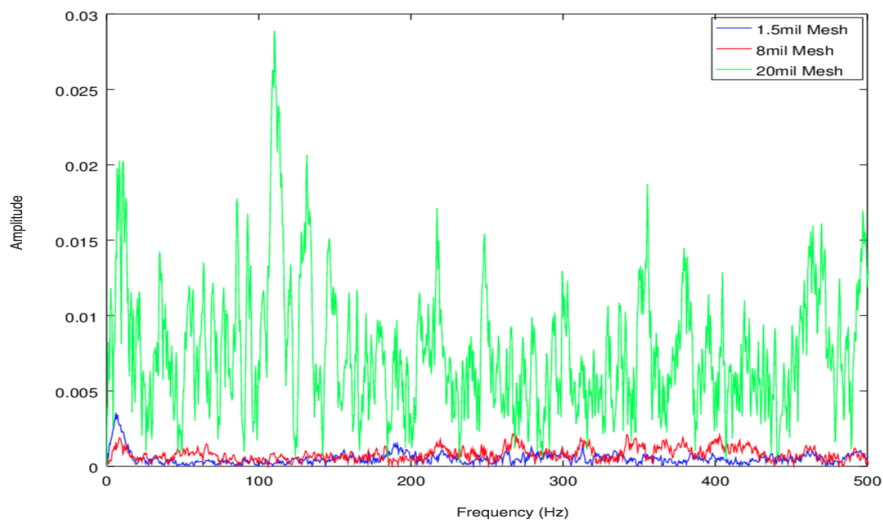
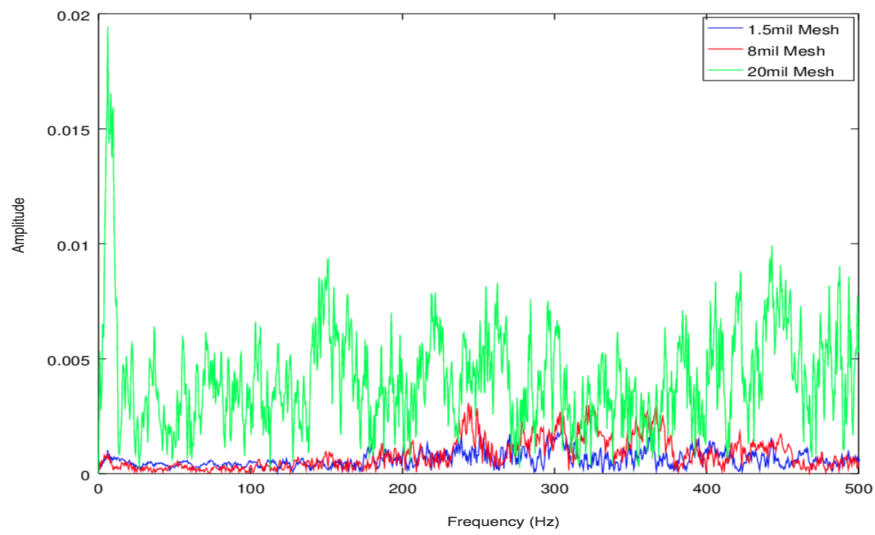
Figure 4.3: Plots over lines for velocity magnitude, mean and variance for channel flow at 900mm height. The plots represent lines 1 to 5 from top to bottom in the same order.

On line 1, it can be seen that the flow space is restricted enough so that the outer layer is not observable. Lines 2 to 4 exhibit all three layers while line 5 has enough space away from the wall that the buffer layer is "smoothed" out and is not immediately visible.

4.1.2 Spectral Analysis

Next, the behavior of the simulations data values is going to be examined in the frequency domain. In figures 4.4 the velocity spectra can be seen for the x component of the turbulent velocity at a height of 900mm inside the computational domain for all 3 computational meshes and all different probe locations, as explained in chapter 3. The first plot corresponds to the probe on the corner of the fuel bundles, the second is opposite of the first, next to the housing tube, the third belongs to the probe next to the tube, on the level of the flow channel and the rest are in the middle of one branch of the flow channel, at increasing distance from the housing tube, the last one being close to the edge of the geometry at the maximum possible distance from the tube.





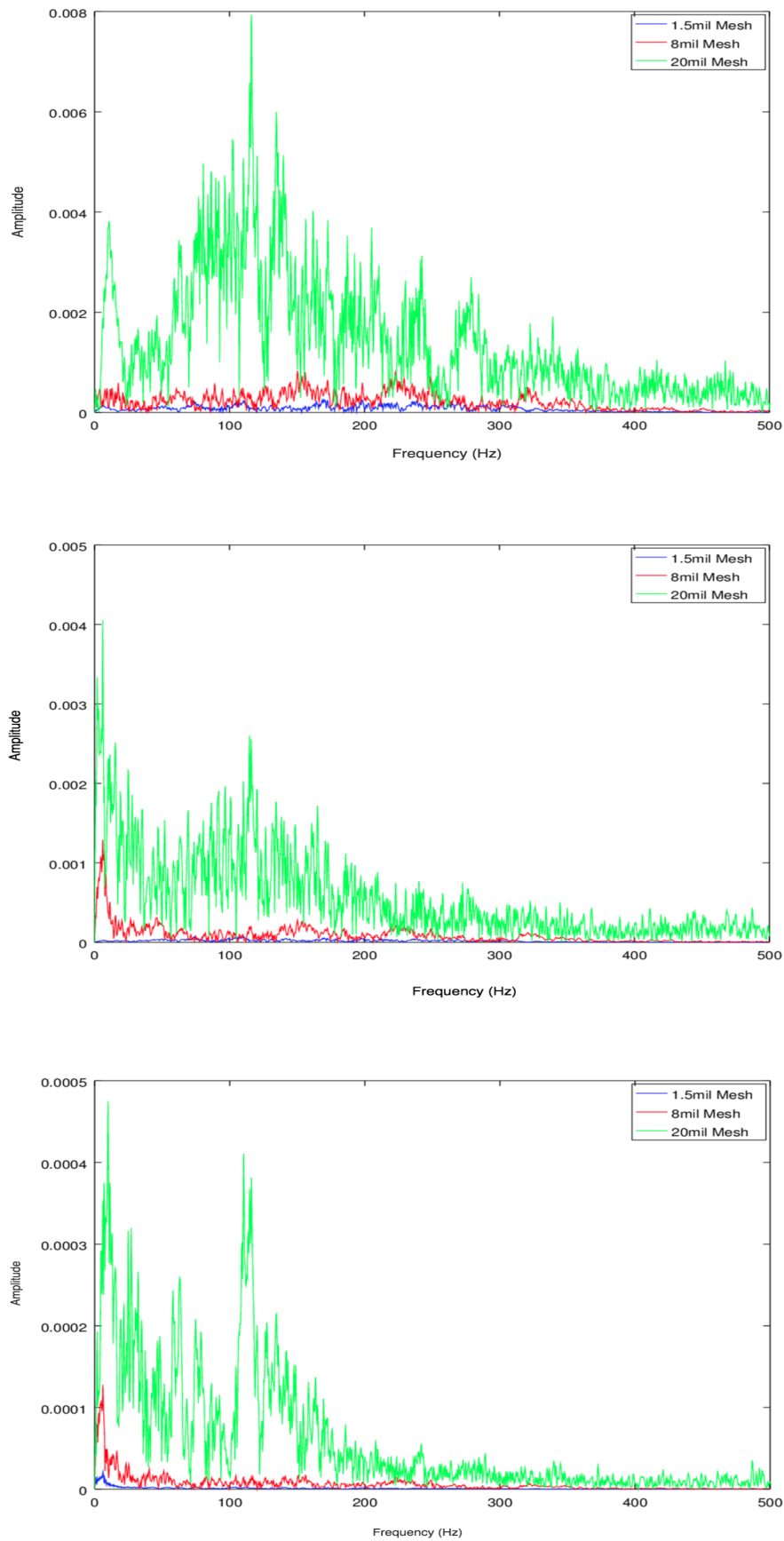


Figure 4.4: Frequency spectra for turbulent velocity's x component for all probes and all computational meshes at a height of 500mm.

It can be seen that there is some correspondence between the turbulent velocity's frequencies and the natural frequencies of the tube, especially on higher (4th and higher) modes for the smaller meshes and the 1st mode away from the narrow flow channel between the tube and the fuel rods. Meanwhile on the high resolution mesh, the turbulent effects are strongly detailed, making it harder to recognize the real peaks in the graphs, but it can be said that there is some characteristic peak in all graphs in the 10Hz region, corresponding to the 1st natural mode, which is the dominating mode on turbulent buffeting. Therefore, it is plausible that FIV due to turbulent buffeting takes place, considering the fact that as it can be seen on figure 4.1 there exist mean velocity flow perturbations, which is the required criterion for such vibrations to take place. The data do not show some characteristic behavior to conclude with certainty that this is the case or the flow - periodicity resonance is present.

Still, fluid-elastic instabilities also appear to be a possibility, since the dimensionless velocity u^+ along the tube's surface appears to be higher than the suggested threshold of 1.5 required for such FIV [3]. The calculations that took place followed Karman's law of the wall 4.3, according to which you can obtain an approximation for the dimensionless flow velocity u^+ by using the dimensionless wall distance y^+ which was calculated to be around 0.55 for the turbulent case and von Karman's constant k which is universally accepted to have a value of 0.4 and C^+ is a positive constant which has a value of approximately 5 for smooth walls. Using these equations and the data produced by the simulation, a dimensionless velocity of at least $u^+ \approx 4$ was calculated which is high enough for fluid instabilities to be present.

The figures were taken at the height of 500mm, corresponding to a nodal (normalized) position of $X_n = 500/1486 = 0.336$. The nodal positions can be calculated from the differential equation for the bending of a slender beam, following the same idea as the calculation of the natural frequencies and then solving for the equations for the eigenfunctions to be equal to zero, to find the nodal positions. By doing that, it can be seen that our calculations are near the 3rd and 6th nodal positions, meaning we will see less response from those modes on the frequency spectra.

The y component spectra are irrelevant for these probe location, because they were all placed in a branch of the flow channel, spanning across the x direction, giving no relevant and significant vibrations along the y axis. Finally, a logarithmic plot showing the energy spectrum over wavenumber can be seen on figure 4.5. The turbulent layers can be observed here as well, especially the logarithmic $k^{-5/3}$ which can be seen as a declining linear relation in the logarithmic plot.

4.1.3 Matrix solver performance

Optimizing included all the available tools at our disposal, mainly controlling the matrix solvers and schemes used for the computations. It was quickly shown that the choice of computational schemes for the different operations made little to no difference, leaving most of the optimization to an examination of the parallel behavior of the different matrix solvers available.

The available matrix solvers included the popular, current state of the art **GAMG**, or generalised geometric-algebraic multi-grid solver, the preconditioned (bi-) conjugate gradient solver (**PCG** or **PBiCG**) for symmetric and asymmetric matrices respectively, as well as a smooth (using a smoother) and diagonal solvers. Each solver is generalized with the ability to have preconditioner and/or a smoother for each solver. The preconditioners

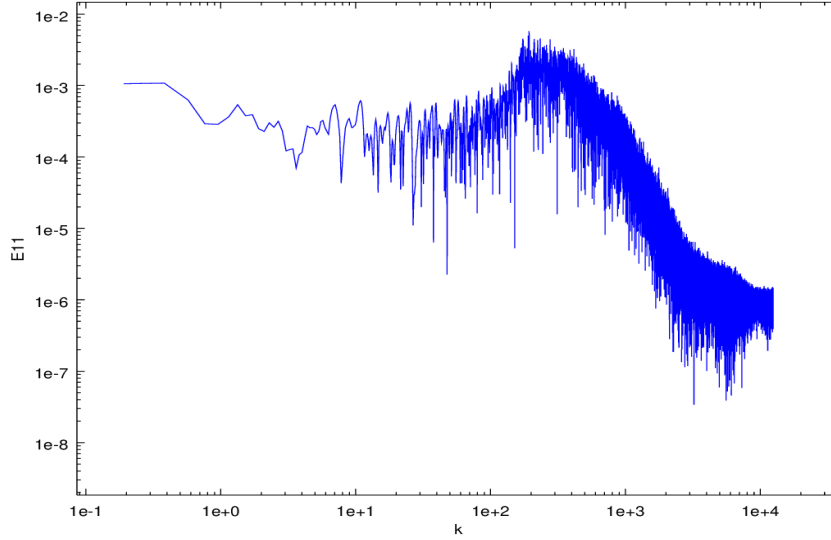


Figure 4.5: Energy spectrum, calculated from the turbulent velocity on logarithmic axis over wavenumber.

include the diagonal incomplete-Cholesky (DIC) and diagonal incomplete LU (DILU), and a GAMG preconditioner. The smoothers include DIC, DILU and Gauss-Seidel; variants of DIC and DILU exist with additional Gauss-Seidel smoothing.

After a little profiling, it became obvious that the heavy computation part was the solution of the pressure equations but the velocity and kinetic energy computation appeared important as well. To show how these solvers behave in general with an increased number of computational resources, various tests have been performed for the turbulent case and the results are presented in table 4.1. The names of the solvers are always listed in pairs, the first corresponding to the pressure equations and the second one to the velocity and kinetic energy equations.

It can be understood from this table that even though the *GAMG* solver is often the optimal choice, for serial applications or small parallel solutions, for larger systems it immediately became apparent that the conjugate gradient solvers are much more efficient and scale much better. The smooth solver for the velocity scales also quite well, and has provided better results than other solvers, as it has been shown from the turbulence study. It is, however, quite unstable when applied to the pressure equations and thus not a good choice for them. It appears from these results that the optimal choice, after all, is always the multi-grid solver considering that even with worse scaling the time required for the fastest execution time in all solver pairs is comparable, making the more accurate and less resource-hungry multi-grid solver the matrix solver of choice, often coupled with a simple iterative smooth solver from the velocity and kinetic energy.

The results summarized in tables 4.1 and 4.2 can be seen in graphical form on figure 4.6 for cells per core, instead of the number of cores. For each solver pair, it can be seen that there is a characteristic point of cells/core below which the solver stops scaling with increasing number of cores but actually becomes slower. A special setting for the multi-grid solver, which allows for processor agglomeration for the coarsest levels, thus allowing the coarsest grids to be solved directly or iteratively on the master processor only and then distributed to the rest of the processors, supposedly limiting communications and improving performance. For our case, however, it immediately becomes apparent

cores	GAMG+smooth	GAMG+GAMG	GAMG+smooth(p.Aggl)	GAMG+GAMG(p.Aggl)
32	346	351	344	353
64	162	168	164	168
128	78	81	83	85
256	52	47	62	63
320	42	44	51	55
384	44	52	92	100
448	42	48	88	94
512	42	46	84	90
640	73	80	91	99
768	106	81	255	269
1024	180	191	455	467
1152	270	222	556	565
1280	244	296	1652	1679

Table 4.1: *Table of time (in seconds) required for 100 steps of a developed flow in the turbulence case for different matrix solvers. Solvers are in pairs of pressure-velocity/kinetic energy.*

that this choice has comparable results on fewer processors but immediately becomes slower, as the scaling increases. This is mainly because the geometry studied is simple and can be easily distributed to processors by cutting it in very thin slices along the tube's axis, making communication less important than calculation and processor agglomeration pointless. These results have been omitted on the graphical representation since they were not of much interest after all.

4.2 FSI simulation

The final step of this chapter, examines the results obtained from the FSI enabled simulation. Making this setup work was more challenging than expected, mainly because of issues with the software. *Foam-extend*, being a community-driven software had many minor errors and was lacking proper documentation. After a little work, a running simulation was able to be created. It was concluded though, that despite the fact that it is a nice initiative, with perspective and many possibilities, *foam-extend* is not ready yet for high performance computing simulations. It lacked severely in the parallelization and optimization part, struggling to find a balance in performance between the two solvers, and achieving a realizable in terms of time-consumption result. It is quite possible that the geometry of this study was not perfect for the solvers available, so as to not be unfair against the software and it might work better for different kind of geometries. The main issue was finding a way to make the simulation realizable in the limited time window available for the work of this project. Even with all the resources available, getting the proper results was calculated to require a whole 2 months of constant computations for a detailed computational mesh. After applying all the possible optimizations at our disposal and limiting the computational mesh to a medium resolution one, a realizable computation time of approximately 20 days for 1 second of simulation time was able to be achieved.

cores	PCG+smooth	PCG+PBiCG
32	1602	1512
64	634	750
128	260	273
256	95	121
320	73	85
384	64	74
448	52	61
512	42	53
576	43	52
768	37	45
1024	29	33
1152	37	36
1280	35	28
1536	31	29

Table 4.2: Table of time (in seconds) required for 100 steps of a developed flow in the turbulence case for different matrix solvers. Solvers are in pairs of pressure-velocity/kinetic energy.

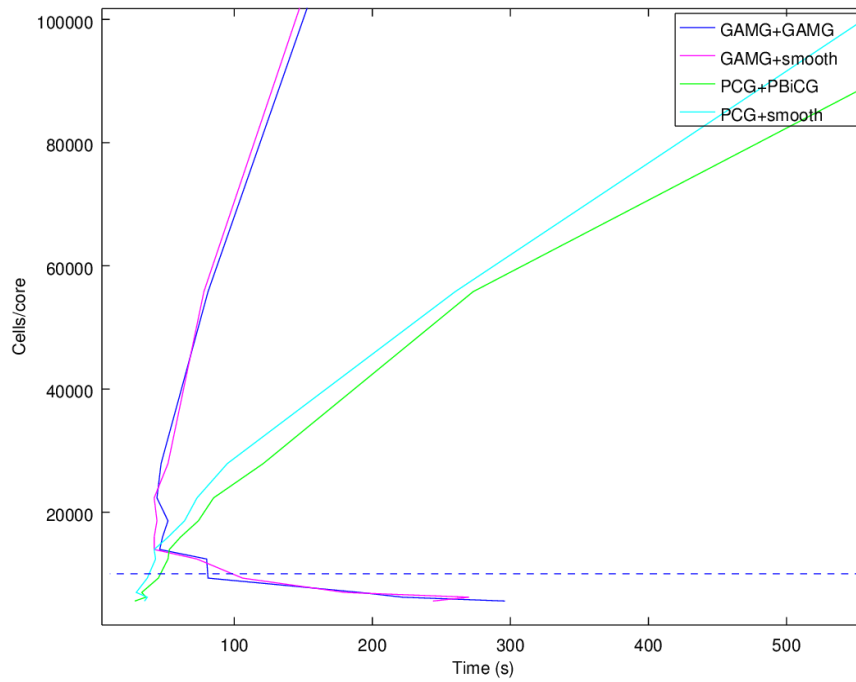


Figure 4.6: Time over cells/core for different matrix solvers.

4.2.1 Performance

Although, as it can be seen, we can highly improve the computation time required for the turbulent case simply by choosing the correct matrix solver, things are not so straightforward with the FSI case. The problem is the imbalance between the size of the fluid and structural meshes. A size of approximately 4.2 million cells for the fluid mesh is quite big, compared to the roughly 250 thousand cells of the structural mesh. Due to the form of the solvers, both meshes need to be split in an equal number of pieces, before distributed to the different cores. This creates the problem that although the fluid mesh can scale to quite a lot cores (the PCG solver scales down to approximately 5580 cells/core), for the same amount of cores the structural mesh will have approximately 300 cells/core. Therefore, with increasing number of cores the computational time does not only stop scaling, but it actually becomes increasingly slower, making it necessary to find a balance between the two meshes.

After some testing, it has been shown that the best results are on 160 cores (5 nodes) with approximately 26000 cells/core for the fluid mesh and 1500-1600 cells/core for the structural mesh. These results have been taken with the multigrid solver, since the conjugate gradient solver appears to be slower and more inaccurate, leading to instabilities, making it inefficient for FSI simulations. This is far from a perfect solution, but it is a better compromise, to make it possible to carry out the computations in a feasible time window.

Furthermore, choosing the optimal time step can be an issue in this situation. Choosing a high enough time step can cause instability due to the CFL condition being violated. On the other hand, a too low time step can cause issues as well due to the so called artificial mass effect [20], which can induce unphysical pressure oscillations beneath the movable part of the structure with decreasing time step size, which are accepted as a valid solution, but could lead to faulty structural displacements and due to the limited space available for the movement of the tube, render the solver incapable to continue working. The perfect balance for this situation appeared to be a timestep of 2×10^{-5} seconds.

A simple profiling function was compiled into the source files of the solver and allowed to run for a complete program loop of approximately 100 steps, which is the time required for 1 write and 1 interpolation update, in order to understand how the time spent was distributed along the program's processes. The data are summarized in table 4.3.

It can be seen from table 4.3 that the time required for solving the fluid equations, which uses actually the same solver as the previously examined turbulent case, requires only approximately 20% of the total main loop. We can realize that the biggest bottlenecks in our simulation's efficiency is the mesh movement (fluid mesh + displacement update) as well as the stress solver, each of which takes up over 30% of the total simulation time.

4.2.2 Mesh Displacement

A minor mistake, which was discovered late, made it necessary to restart the process since it produced unnatural behavior which caused the computational mesh to distort excessively and non-orthogonality issues to appear and cause the simulation to stall. Thus the results were taken from a second run which was allowed to develop for approximately half a second, with a sampling rate of $20 \mu s$ which was deemed sufficient for a complete and proper analysis. On figure 4.7 the displacement on a surface vertical to the flow normal direction can be seen which underlines the fact that the amplitude of the vibrations is extremely small, however significant when it comes to quality assurance. Another inter-

Process	Time
Main Loop	4438.9 sec
Data initialization	0.0351438 (%)
Interpolation update	3.0494 (%)
Displacement update	11.4197 (%)
Fluid mesh update	22.9352 (%)
Flow solver	19.7578 (%)
Force update	8.56113 (%)
Stress (solid) solver	32.2219 (%)
Residual update	1.96332 (%)
Write data	0.056095 (%)

Table 4.3: *Profiling data for a complete loop of the FSI case (100 steps).*

esting effect, as shown in figure 4.8 is the vibration of the tube in the flow direction, due to the pinned upper boundary condition, which shows a high frequency declining forced vibration which appears in bursts at fixed intervals and increasingly further of the tube which could suggest that the rod is either bending or is being compressed along the flow direction, with the former being a more logical explanation and reinforcing the evidence pointing to FIV due to fluttering/bending from high velocity flow along with a later examination of the history data of the probe locations on the computational grid that showed that the tube was, indeed, bent at a height of approximately 950mm, which is normalized to $950/1486 = 0.6393$ which is extremely close to the 3rd and 5th modes of the tube. An examination of possible ways to decrease these effects is necessary in such a work. Possible solutions are decreasing the flow velocity or increasing the rigidity of the tube. In case both of these are impossible, increasing the tension of the tube on the downstream end or increasing the tube's length have both been proposed as valid reasons for decreasing the effect of fluttering/bending [31].

4.2.3 Spectral Analysis

Similar to the turbulent case, an analysis of the computational data in the frequency domain has been performed. In this case, the velocity data of the structural mesh have been examined instead, in order to see their correspondence with the experimental data which involved the point velocity on the structural domain. On figures 4.9 through 4.13 plots of the superimposed frequency data at a point on the tube near its edge and at different heights, specifically at 350mm, 550mm, 750mm, 950mm and 1150mm from the bottom end of the tube can be seen for the x and y velocities.

The spectra differ from each other due to the fact that each data collection point is close to a different nodal position(s) of the natural frequencies of the tube, thus seeing less response from said frequency in the respective spectrum. As explained earlier in the spectral analysis of the turbulent case that is similar to this one, we can pinpoint an approximate correspondence between the peaks in the graphs and the theoretically calculated natural frequencies of the housing tube with total or partial absence of the frequencies whose nodal positions are close to the point where the data are collected.

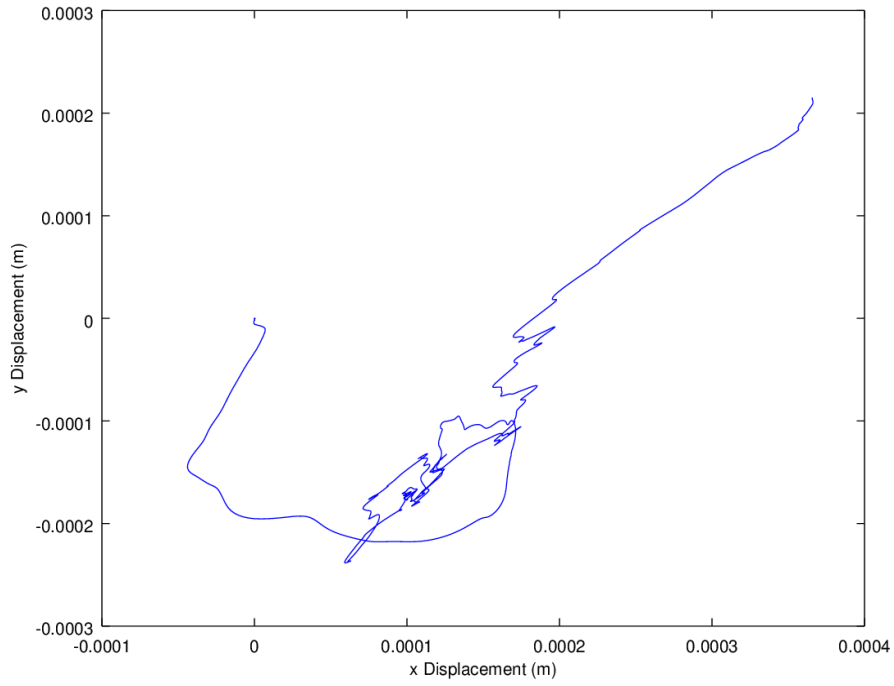


Figure 4.7: Displacement of a point on the structural mesh in a surface vertical to the flow normal direction.

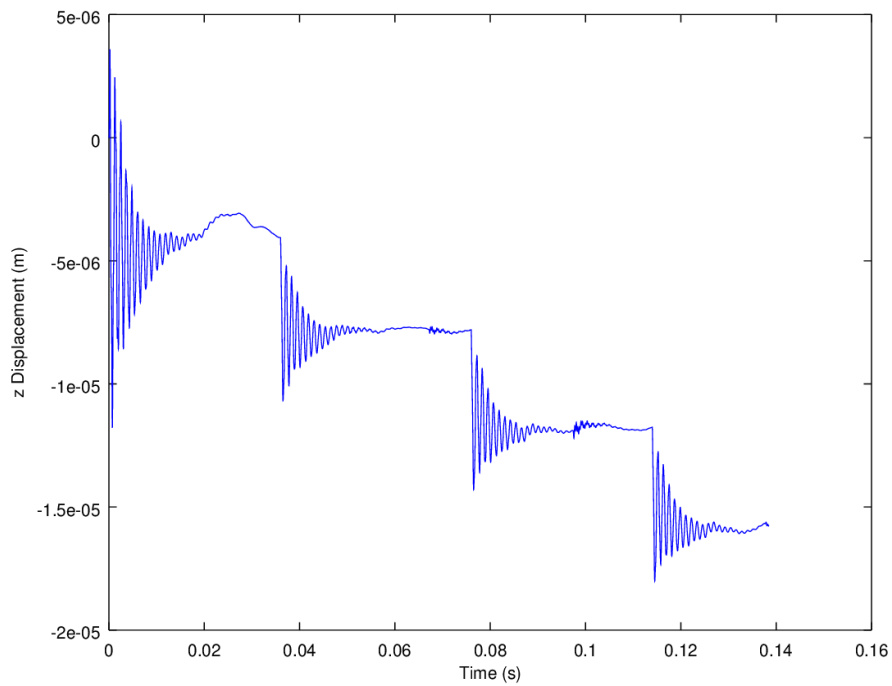


Figure 4.8: Displacement of a point on the structural mesh over time in the flow normal direction.

Additionally, on figure 4.14 a spectrum for the tube's x axis movement for a stationary boundary point of the tube at a height of 950mm and on figure 4.15 the spectrum of the pressure along the x-axis on the tube's surface can be observed.

Finally, on figure 4.16 we can observe the spectra obtained from the experimental setup for different inlet mass flows (the 10 kg/m^3 mass flow used in the simulation is represented by the red line) and different data sets, including velocity, position, acceleration etc. The data were collected at a height of 900mm from the core's grid base so the velocity data should be close with the velocity frequency spectrum calculated from the simulation at a height of 950mm. The data represented here are restrict to the x-axis only. From the data collected and presented, we can deem the attempt for a successful simulation of the wanted geometry as successful, considering the close resemblance of the experimental and the calculated results.

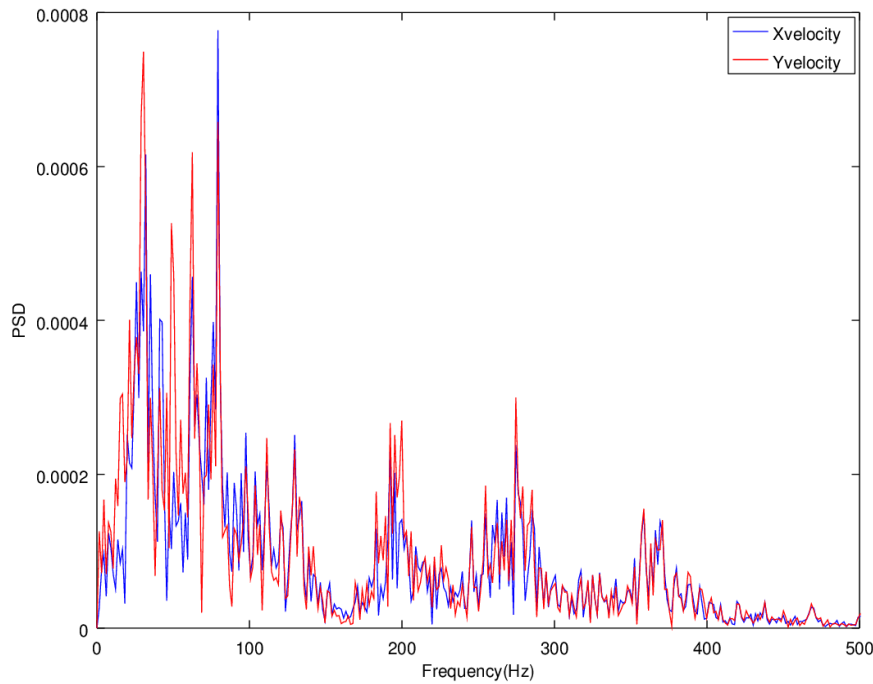


Figure 4.9: Spectrum of the housing tube's x and y velocities at a height of 350mm.

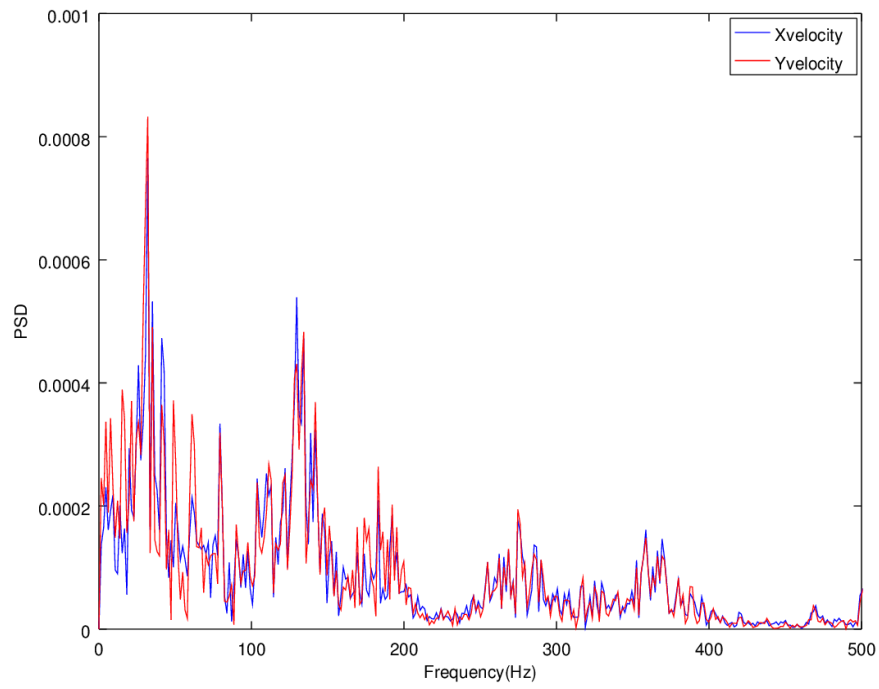


Figure 4.10: Spectrum of the housing tube's x and y velocities at a height of 550mm.

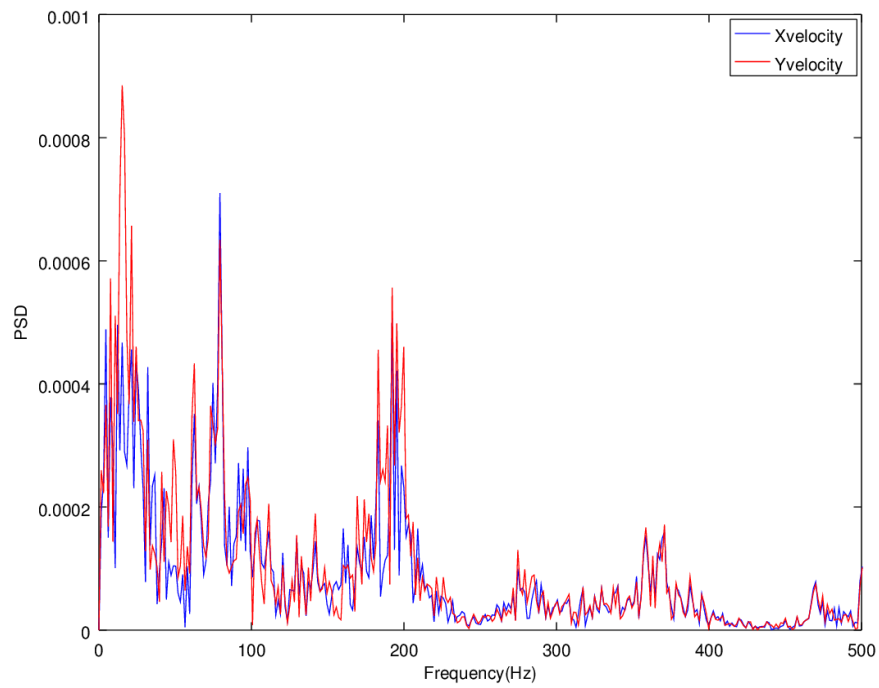


Figure 4.11: Spectrum of the housing tube's x and y velocities at a height of 750mm.

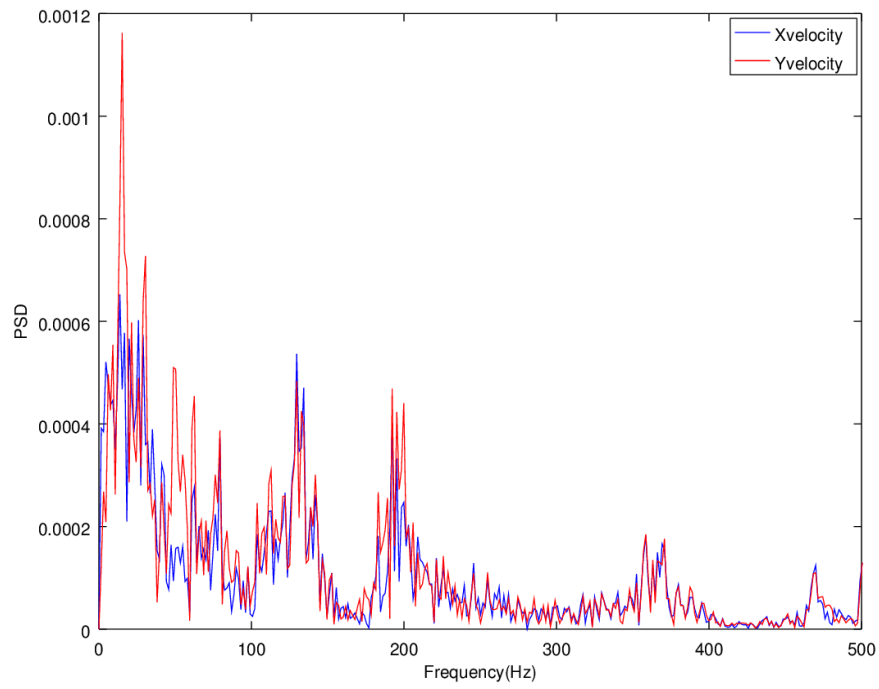


Figure 4.12: Spectrum of the housing tube's x and y velocities at a height of 950mm.

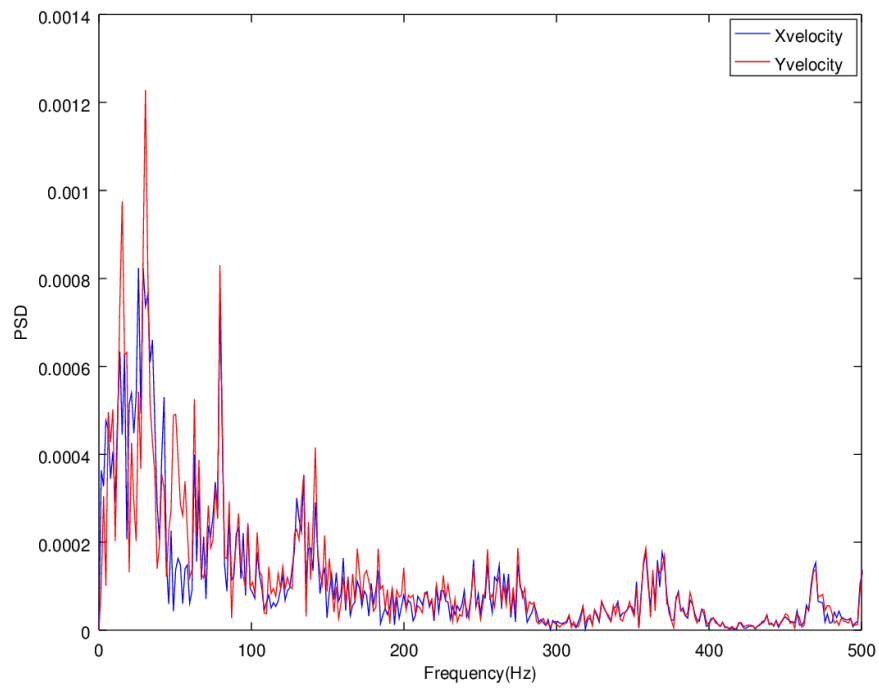


Figure 4.13: Spectrum of the housing tube's x and y velocities at a height of 1150mm.

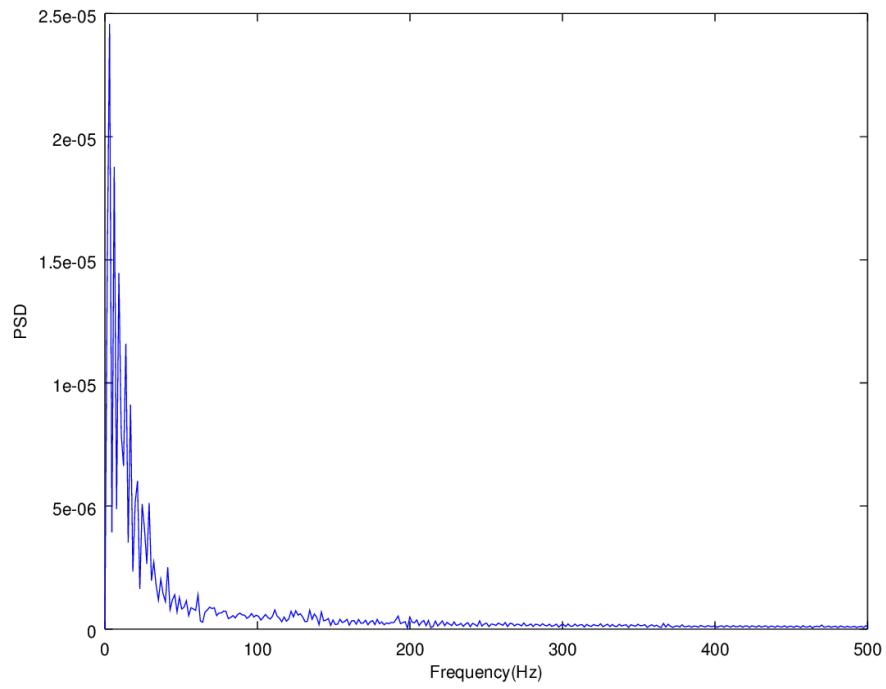


Figure 4.14: Spectrum of the housing tube's x axis displacement at a height of 950mm.

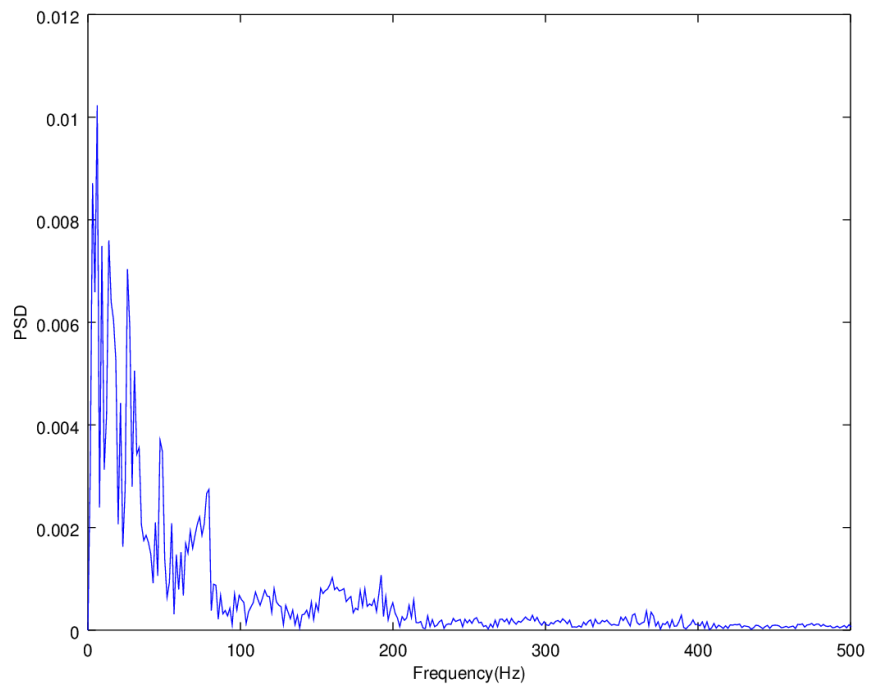


Figure 4.15: Spectrum of the pressure along the tube's surface for the x component.

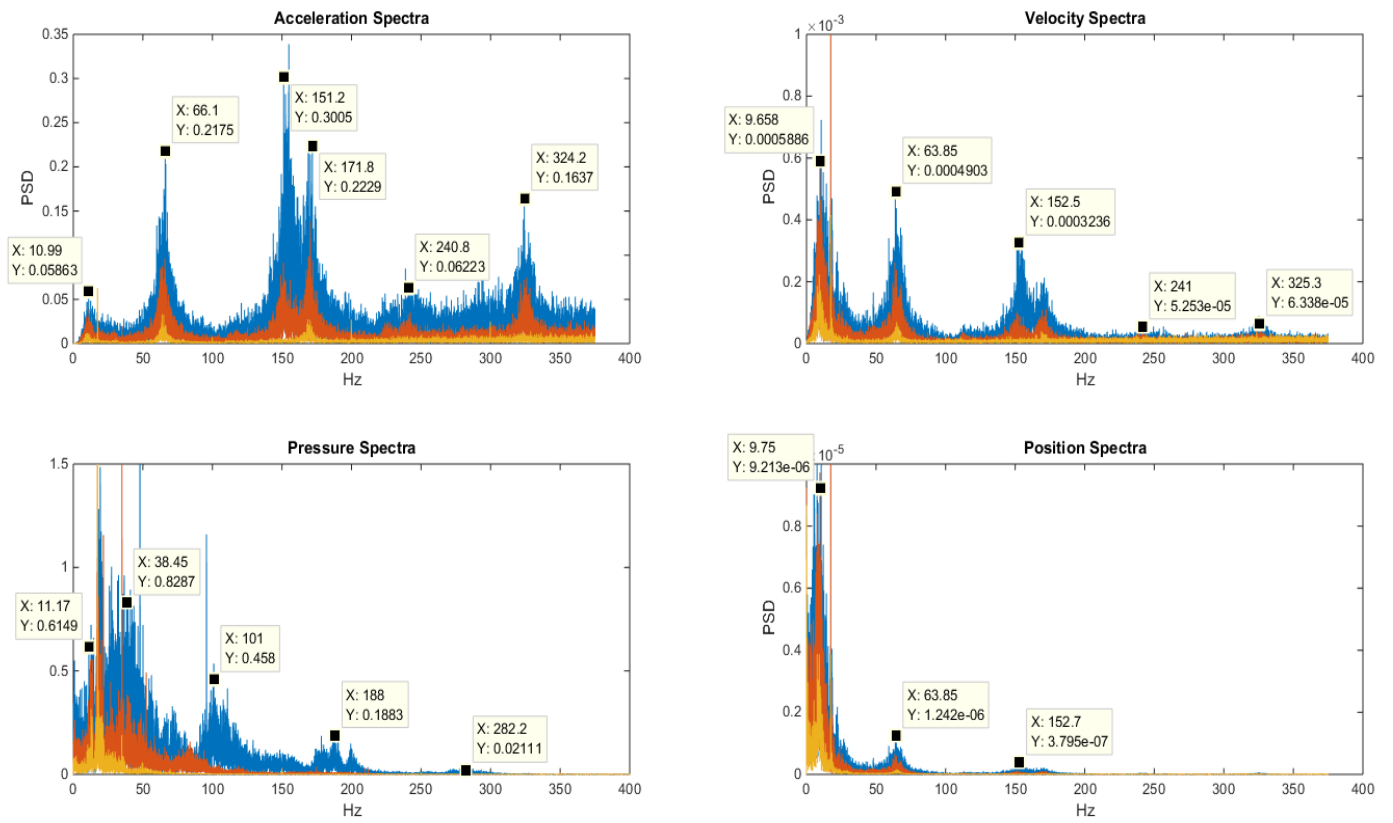


Figure 4.16: Frequency spectra for experimental setup (provided by the project's supervisor[1])

5. Future work & Conclusion

This thesis work presented a validation of an industrial application and experiment with the use of open-source CFD package *OpenFOAM*.

Specifically an axial flow induced vibration model was proposed for a rod fixed on the grid at the lower end and supported by a spring (pinned) on the upper end. The natural frequencies and bending modes of the tube were calculated by solving the eigen-problem for a fixed-pinned beam. The vibrations were calculated through the simulation for a rod with a length of 1486 mm subjected to a flow mass rate of 10 kg/m^3 as well as the corresponding turbulent model, for varying flow mass rates.

Additionally, this work was designed to act as a benchmark for similar applications in regards to the geometry, solver or the CFD software *OpenFOAM*. The Fluid-Structure Interaction solver *FsiFoam* was used, along with the *PISOFoam* solver for the turbulent case. *FsiFoam* contains a specialized form of the *PISOFoam* solver for the fluid domain along with the FVM *stressFoam* for the solid domain.

The validation of the experimental setup involved:

- Statistical representation of the results.
- Spectral analysis of both the turbulent and FSI simulations.
- Performance investigation for the matrix solvers, numerical schemes, mesh motion handling and other less significant aspects of the simulations.

A reproduction of the experimental data in terms of the amplitude of the FIV as well as the frequency content of said vibrations was achieved for the FSI simulation, proving the capability of the CFD package *OpenFOAM* to provide correct and accurate coupled fluid and structural simulations. The investigation of the turbulent case for the cause of the FIV was not as successful. Only partial correspondence to the tube's natural frequencies were present and only on the first and higher than 4th modes, suggesting that the FIV should not have been caused by flow periodical resonance. However, the possibility for turbulent-buffeting, which is one of the main expected reasons for such vibrations [7], is present. However the calculated dimensionless velocity appeared to be high enough which suggested the existence of flow-instabilities like buckling and fluttering due to the turbulent flow. A future work could search and examine other possible excitation sources although the most widely accepted causes have already been examined. An examination of possible ways to decrease these issues would also be useful.

A slight error in the experimental setup was discovered later, causing a misjudge in the elasticity modulus chosen for the problem in the water domain but should not alter the expected results. This was discovered due to the efforts of a similar work, with the same setup but a different CFD package called *ANSYS* that was performed approximately at the same time as this work at Chalmers University of Technology and would be interesting for a reader of this work.

Another possible future examination of this analysis would be to perform the same simulation for a bigger, more accurate and detailed mesh. This was not achievable in the time frame of the current work and the original mesh had to be limited in order to get results in the expected time frame. Examination of different CFD packages would also be important to consider as an alternative, to increase the acceptance of FSI simulations as valid industrial analysis of such problems. Finally, in regards to the HPC capability of

the software, a more detailed and in depth profiling of the solvers and utilities used could be performed.

References

- [1] E. Lillberg, K. Angele, G. Lundqvist, and N. Edh, “Tailored experiments for validation of CFD with FSI for nuclear applications.” Unpublished Manuscript, 2015.
- [2] K. Y. Billah and R. H. Scanlan, “Resonance, Tacoma Narrows bridge failure, and undergraduate physics textbooks,” *Am. J. Phys*, vol. 59, no. 2, pp. 118–124, 1991.
- [3] M. Paidoussis, “A review of flow-induced vibrations in reactors and reactor components,” *Nuclear Engineering and Design*, vol. 74, no. 1, pp. 31–60, 1983.
- [4] P. Nilsson and E. Lillberg, “Risks for Flow Induced Vibrations (FIV) at EPU,” in *16th International Conference on Nuclear Engineering*, pp. 237–244, American Society of Mechanical Engineers, 2008.
- [5] M. P. Paidoussis, *Fluid-structure interactions: slender structures and axial flow*, vol. 1. Academic press, 1998.
- [6] M. Paidoussis, “Dynamics of flexible slender cylinders in axial flow Part 1. Theory,” *Journal of Fluid Mechanics*, vol. 26, no. 04, pp. 717–736, 1966.
- [7] M. Paidoussis, “Fluidelastic vibration of cylinder arrays in axial and cross flow: state of the art,” *Journal of Sound and Vibration*, vol. 76, no. 3, pp. 329–360, 1981.
- [8] ESI-OpenCFD, “OpenFOAM, The Open Source CFD Toolbox.” <http://www.openfoam.com/>, May 2015.
- [9] “The OpenFOAM Extend Project.” <http://www.extend-project.de/>, June 2015.
- [10] SALOME, “The Open Source Integration Platform for Numerical Simulation.” <http://www.salome-platform.org/>, April 2015.
- [11] “GNU Octave.” <https://www.gnu.org/software/octave/>, August 2015.
- [12] “PDC center for High Performance Computing.” <https://www.pdc.kth.se>, April 2015.
- [13] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics*. Springer Science & Business Media, 2012.
- [14] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.
- [15] S. B. Pope, *Turbulent flows*. Cambridge university press, 2000.
- [16] J. Taghinia, M. M. Rahman, T. Siikonen, and R. K. Agarwal, “One-equation sub-grid scale model with variable eddy-viscosity coefficient,” *Computers & Fluids*, vol. 107, pp. 155–164, 2015.

- [17] J. Degroote, K.-J. Bathe, and J. Vierendeels, “Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction,” *Computers & Structures*, vol. 87, no. 11, pp. 793–801, 2009.
- [18] H. Jasak and Z. Tukovic, “Dynamic mesh handling in openfoam applied to fluid-structure interaction simulations,” in *Proceedings of the V European Conference Computational Fluid Dynamics, Lisbon, Portugal, June*, pp. 14–17, 2010.
- [19] J. Degroote, R. Haelterman, S. Annerel, A. Swillens, P. Segers, and J. Vierendeels, “An interface quasi-newton algorithm for partitioned simulation of fluid-structure interaction,” in *International Workshop on Fluid-Structure Interaction. Theory, Numerics and Applications*, p. 55, kassel university press GmbH, 2009.
- [20] H.-J. Bungartz, M. Mehl, and M. Schäfer, *Fluid Structure Interaction II: Modelling, Simulation, Optimization*, vol. 73. Springer Science & Business Media, 2010.
- [21] Yigit, Stenel, and Schäfer, “Efficiency of fluid-structure interaction simulations with adaptive underrelaxation and multigrid acceleration,” *The International Journal of Multiphysics*, vol. 1, no. 1, pp. 85–99, 2007.
- [22] M. Souli, A. Ouahsine, and L. Lewin, “ALE formulation for fluid-structure interaction problems,” *Computer methods in applied mechanics and engineering*, vol. 190, no. 5, pp. 659–675, 2000.
- [23] F. Liou, J. Newkirk, Z. Fan, T. Sparks, X. Chen, K. Fletcher, J. Zhang, Y. Zhang, K. S. Kumar, and S. Karnati, “Multiscale and multiphysics modeling of additive manufacturing of advanced materials,” 2015.
- [24] S. Krenk, *Mechanics and analysis of beams, columns and cables: a modern introduction to the classic theories*. Springer Science & Business Media, 2001.
- [25] CFD Direct. <http://cfd.direct/openfoam/user-guide/snappyhexmesh/>, May 2015.
- [26] H. Jasak, A. Jemcov, and Z. Tukovic, “OpenFOAM: A C++ library for complex physics simulations,” in *International workshop on coupled methods in numerical dynamics*, vol. 1000, pp. 1–20, 2007.
- [27] R. I. Issa, “Solution of the implicitly discretised fluid flow equations by operator-splitting,” *Journal of computational physics*, vol. 62, no. 1, pp. 40–65, 1986.
- [28] W. Thomson, *Theory of vibration with applications*. CRC Press, 1996.
- [29] I. A. Bolotnov, R. T. Lahey, D. A. Drew, K. E. Jansen, and A. A. Oberai, “Spectral analysis of turbulence based on the DNS of a channel flow,” *Computers & Fluids*, vol. 39, no. 4, pp. 640–655, 2010.
- [30] R. D. Blevins, *Flow-induced vibration*. New York, NY (USA); Van Nostrand Reinhold Co., Inc., 1990.
- [31] T. Nakamura, S. Kaneko, F. Inada, M. Kato, K. Ishihara, T. Nishihara, and N. W. Mureithi, *Flow-induced vibrations: Classifications and lessons from practical experiences*. Butterworth-Heinemann, 2013.

Stockholm University
Department of Physics
Stockholm University
SE-106 91 Stockholm
Phone: +46 8 5537 8000
www.fysik.su.se



Stockholm
University

HPC FSI SIMULATION OF A ROD SUBJECTED TO AXIAL TURBULENT FLOW

The study of flow-induced vibrations on slender structures is a relatively new subject, with increasing interest in recent years. In the present work the axial turbulent flow past a cylindrical object is simulated, by using an open source CFD code (OpenFOAM) with a Large Eddy Simulation (LES) approach. High Performance Computing (HPC) resources were available and a detailed approach has been accomplished, with several computational meshes scaling up to 20 million cells. Spectral and spatial analysis has been performed on the numerical results and validation with the experimental data provided has been carried out.

Additionally, an FSI simulation has been achieved for the same geometry, with a partitioned approach, coupling two separate flow and structural solvers and a comparison has been carried out with the experiment here as well, in order to examine the ability of the software to predict the behavior of the FSI and how the numerical approximations affect the results in turbulent cases.

The turbulent velocity frequencies showed a good correspondence with the natural vibration modes of the rod and the software was shown capable to simulate the FSI behavior, but had room for improvement in several sectors, especially HPC. Many possible causes have been considered as possible for the FIV and numerical calculations of the problem's variables has shown the existence of fluid-elastic instabilities as the most probable cause of vibrations in the current setup.

Another step forward in Swedish energy research

Energiforsk – Swedish Energy Research Centre is a research and knowledge based organization that brings together large parts of Swedish research and development on energy. The goal is to increase the efficiency and implementation of scientific results to meet future challenges in the energy sector. We work in a number of research areas such as hydropower, energy gases and liquid automotive fuels, fuel based combined heat and power generation, and energy management in the forest industry. Our mission also includes the generation of knowledge about resource-efficient sourcing of energy in an overall perspective, via its transformation and transmission to its end-use. Read more: www.energiforsk.se



Energiforsk