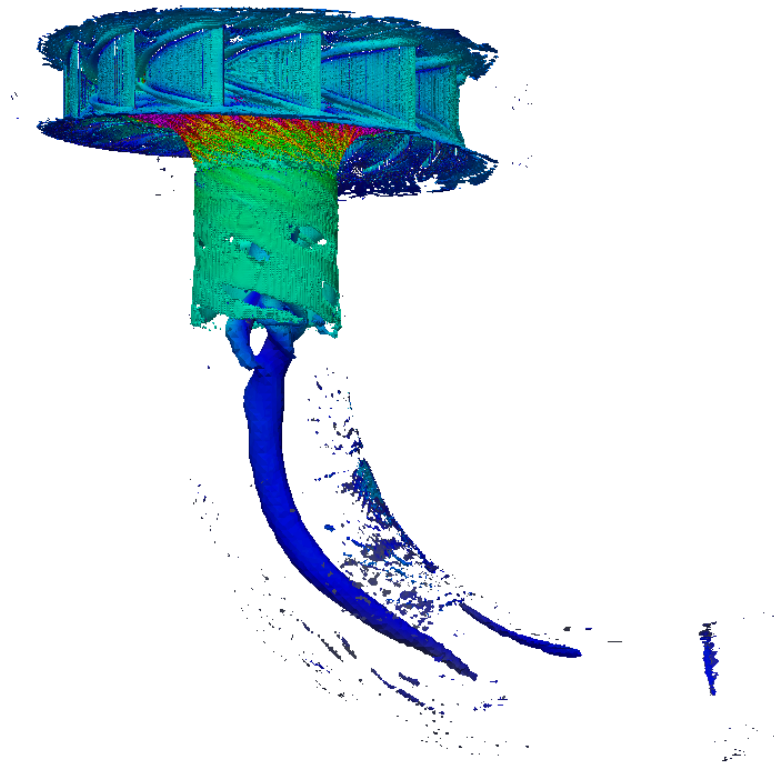




CHALMERS
UNIVERSITY OF TECHNOLOGY



LES of Transients in the Francis-99 Turbine

Numerical Simulation and Verification of Shutdown and Start-up Procedure for the Francis-99 Turbine Using OpenFOAM

Master's thesis in Applied Mechanics

JONATHAN FAHLBECK
LUDVIG UPPSTRÖM

MASTER'S THESIS 2018:17

LES of Transients in the Francis-99 Turbine

Numerical Simulation and Verification of Shutdown and Start-up
Procedure for the Francis-99 Turbine Using OpenFOAM

JONATHAN FAHLBECK
LUDVIG UPPSTRÖM



Department of Mechanics and Maritime Sciences
Division of Fluid Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

LES of Transients in the Francis-99 Turbine
Numerical Simulation and Verification of Shutdown and Start-up Procedure for the
Francis-99 Turbine Using OpenFOAM
JONATHAN FAHLBECK, LUDVIG UPPSTRÖM

© JONATHAN FAHLBECK, LUDVIG UPPSTRÖM 2018.

Supervisor: Eric Lillberg, Senior R&D Engineer, Fluid Mechanics, Vattenfall AB
Examiner: Prof. Håkan Nilsson, Mechanics and Maritime Sciences, Chalmers

Master's Thesis 2018:17
Department of Mechanics and Maritime Sciences
Division of Fluid Mechanics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An iso-surface of the Q -criterion with velocity magnitude visualised on a side view of the Francis-99 turbine.

Typeset in L^AT_EX
Gothenburg, Sweden 2018

LES of Transients in the Francis-99 Turbine
Numerical Simulation and Verification of Shutdown and Start-up Procedure for the
Francis-99 Turbine Using OpenFOAM
Jonathan Fahlbeck, Ludvig Uppström
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Most water turbines are designed to only operate at their best efficiency point and to hardly ever change their operation condition. Modern life and new energy sources pushes the demands making the hydro turbines work as stabilizers to the electric grid. Thus, varying its operation point several times a day. The majority of today's water turbines have a confidential design, leading to difficulties for academia and companies to do research. Hence the creation of the Francis-99 workshops. It is an open access study containing both experimental data and a Francis turbine geometry.

This thesis presents CFD simulations of transients, using LES in OpenFOAM, based on the Francis-99 design. An entire shutdown sequence and four short start-up scenarios are simulated and verified against available experimental data. The numerically simulated data matches the experimental to a great extent. Making the simulation strategy highly suitable to perform more comprehensive investigations e.g. to find optimal transients between operation points that reduce stresses on the turbine. It is further concluded that the axial torque varies linearly with the operation point and that the axial thrust is fairly constant. Also found is that the runner exhibits unwanted radial loads during shutdown which, with time, may fatigue the machine.

Keywords: Francis-99 Turbine, LES, Transient, Start-up, Shutdown, OpenFOAM, CFD, Mesh Morphing, Hydro Power, dynOneEqEddy

Preface

This report is the result of a masters thesis work carried out during the spring term of 2018 at the Chalmers University of Technology, department of Mechanics and Maritime Sciences. It was a collaborative project between Vattenfall AB's R&D department in Älvkarleby and Chalmers division of Fluid Mechanics in Gothenburg. The aim of the work was to investigate the possibility to simulate transients in the Francis-99 hydro-turbine using OpenFOAM. The project conclude that OpenFOAM and Large Eddy Simulation is a suitable platform on which to perform further and deeper analysis of transients in Francis hydro turbines.

Acknowledgements

We would firstly like to thank our supervisor Eric Lillberg at Vattenfall and examiner prof. Håkan Nilsson at Chalmers. Without your support and knowledge in OpenFOAM, hydro power and fluid mechanics in general this thesis work would not have been possible to conduct. Post Doctor Ardalan Javadi at Chalmers also deserves a special thanks. This since he in the initial steps of the project helped us with numerical setup and gave hints on how to think when modifying the source code in OpenFOAM.

The Swedish Hydropower Centre (SVC) helped with financial support, SVC thereby earn an extra amount of gratitude. The support enabled the authors to travel, between Vattenfall's R&D department in Älvkarleby and Chalmers in Gothenburg, several times during the project work.

Finally we would like to thank Patrik Andreasson from Vattenfall's R&D department. It was Patrik who first introduced us to the available thesis projects at Vattenfall and in a way made us choose this particular project.

Jonathan Fahlbeck, Ludvig Uppström, June 2018

Contents

Nomenclature	xii
Physical Quantities	xiii
1 Introduction	1
1.1 Background	1
1.2 Aim	2
1.3 Limitations	2
1.4 Specification of issue under investigation	2
2 Theoretical Framework	3
2.1 Water Turbines	3
2.1.1 Francis turbine	3
2.2 Francis-99	4
2.2.1 Predefined Probes and Velocity Lines	4
2.3 Fluid mechanics	6
2.4 Computational Fluid Dynamics	6
2.4.1 Finite Volume Method	7
2.4.2 Large Eddy Simulation	7
2.4.3 The Smagorinsky Model	8
2.4.4 The Dynamic Model	8
2.4.5 The One Equation Eddy Viscosity Model	9
2.5 Mesh Modification	9
2.5.1 Laplacian Smoothing	10
2.6 OpenFOAM	10
3 Methodology	11
3.1 Simulation Procedure and Work Flow	11
3.2 Pre-Processing	11
3.2.1 Turbulence Modeling	11
3.2.2 Solver Settings	12
3.2.3 Pressure and Velocity Coupling	12
3.2.4 Numerical Schemes and Time Step	12
3.2.5 Boundary Conditions	13
3.2.6 Transient Conditions	14
3.2.7 Mesh	14
3.2.7.1 Mesh Motion	15

3.2.7.2	Generate Guide Vane Mesh	16
3.2.7.3	Replace Guide Vane Mesh and Map Fields	16
3.2.7.4	flattenMesh	18
3.3	Post-Processing	18
3.3.1	Probes	18
3.3.2	Fields	18
3.3.3	Q-Criterion	19
3.3.4	Mean and Fluctuating Values and Standard Deviation	19
3.3.5	Forces	19
3.3.6	Data Filter	20
3.3.7	Fast Fourier Transform	20
4	Results and Discussion	21
4.1	Steady at Best Efficiency Point	21
4.1.1	Pressure Probes	21
4.1.2	Velocity Probes	22
4.1.3	Velocity Field	23
4.1.4	Static Pressure Field	24
4.1.5	Comments About the Steady Results	24
4.2	Guide Vane Meshes and Mesh Morphing	24
4.3	Shutdown Transient	26
4.3.1	Pressure Probes	26
4.3.2	Velocity Lines	27
4.3.2.1	Velocity Line 1	28
4.3.2.2	Velocity Line 2	29
4.3.2.3	Velocity Line 3	30
4.3.3	Shaft Thrust and Torque	31
4.3.4	Frequency Spectrum	32
4.3.5	Field Plots	34
4.3.5.1	Velocity Field	34
4.3.5.2	Q-Criterion	35
4.3.6	Comments About the Shutdown Results	35
4.4	Start-Up Transients	36
4.4.1	Pressure Probes	37
4.4.2	Velocity Line 2	37
4.4.3	Shaft Thrust and Torque	38
4.4.4	Frequency Spectrum	39
4.4.5	Field Plots	40
4.4.5.1	Q-Criterion	40
4.4.5.2	Velocity Field	41
4.4.6	Comments About the Start-Up Results	42
5	Conclusions	43
6	Further Work	44
	Bibliography	45

A	Appendix	48
A.1	gv_mesh_script	48
A.2	mergeF99	49
A.3	case_setup	49
	A.3.1 make_frozen_and_rotating_v2	49
	A.3.2 123myFlattenMesh	50
A.4	doublePimpleDyFoam_333rpm	52
A.5	Guide Vane Modified BC	54

Nomenclature

Abbreviation	Meaning
AMI	Arbitrary Mesh Interface
BC	Boundary Conditions
BEP	Best Efficiency Point
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
DT	Draft Tube
EXP	Experiment
FFT	Fast Fourier Transform
FVM	Finite Volume Method
GV	Guide Vane
LES	Large Eddy Simulation
LTU	Luleå Tekniska Univeritet
NTNU	Norges Teknisk-Naturvetenskapliga Universite
PDE	Partial Differential Equation
PIV	Particle Image Velocimetry
PL	Part Load
RANS	Reynolds-Averaged Navier-Stokes
RPM	Revolutions Per Minute
SC	Spiral Casing
SD	Standard Deviation
SGS	Sub Grid Scales
SIM	Simulation
SV	Stay Vane

Physical Quantities

Variable	Symbol	SI-unit
Courant Number (CLF)	Co	-
Density	ρ	kg/m ³
Diameter	d	m
Diffusivity factor	γ	-
Dynamic Leonard stresses	\mathcal{L}_{ij}	kg/(m s ²)
Grid Size/Filter width	Δ	m
Guide vane angle	α	°
Kinematic viscosity	ν	m ² /s
Kronecker delta	δ_{ij}	-
Pressure	p	kg/(m s ²)
Time	t	s
SGS stress tensor	τ_{ij}	kg/(m s ²)
Thrust/Force	F	kg m/s ²
Torque/Moment	M	kg m ² /s ²
Turbulent kinetic energy	k	m ² /s ²
Velocity	u, U	m/s
Volumetric flow rate	\dot{V}	m ³ /s

1

Introduction

1.1 Background

Renewable energy sources are in high demand and it continues to grow. Extracting power from water is an old technique but with modern computational tools, e.g. Computational Fluid Dynamics (CFD), new possibilities have been brought to this branch of energy producing machines. However, the demands of modern life push the requirements and with low profits the need of optimal efficiency and life time are huge [1]. But since most of the hydro turbines in the world are confidential, it is hard for academia to do research. Thus, the founding of the Francis-99 project as a three stage workshop organized by Norges Teknisk-Naturvetenskapliga Universitet (NTNU) [2] and Luleå Tekniska Univeritet (LTU) [3]. It is an open access Francis turbine regarding design and data with the purpose to provide a platform to do research on for people all over the world. The first workshop, held in 2014, treated steady flow through the turbine. The second, 2016, was to investigate transient operations and fields of interest consisted of e.g. start-up and shutdown procedure of the turbine. The third is planned to the end of 2018 and will examine fluid structure interaction.

Vattenfall AB is one of the largest power companies in Europe, owned by the Swedish government. As one of the biggest suppliers of electrical power in Sweden they own both nuclear- and hydro plants as well as a large number of wind turbines. The research and development centre for water turbines is located in the small town of Älvkarleby by Daläven where also one of Sweden's initial hydro plants is situated.

Turbines are generally designed to start and stop just a few times a year and mainly operate at their Best Efficiency Point (BEP). Lately water turbines have been used to stabilize the power grid during periods of high load, e.g. during afternoons, generating varying operation points. These varying conditions cause load cycles way beyond their design point and risk of fatigue and wear out increase [4].

Start-up and shutdown procedures of a water turbine have a decisive impact on runner blade life time, but the optimal procedures are yet unknown since it has not been of much importance. Measurements made on a start-up sequences of a prototype Francis turbine show that shaft torsion correlate well with the stresses on the runner blades [5]. Thus, it may perhaps be possible to optimize transients via correlating pressure fluctuations on the runner blades to shaft thrust and torque.

1.2 Aim

The main objective of this project is to investigate the possibility to numerically simulate start-up and shutdown procedure of the Francis-99 turbine. The work is to be conducted as transient CFD simulations in the open source software OpenFOAM. The simulations are to be compared to experimental data gathered by NTNU at the Water Power Laboratory in Trondheim, Norway. This to validate the accuracy and feasibility. The final outcome of the project is to determine start-up/shutdown procedures which generate the minimum amount of stress on the turbine and thus plausibility the longest life time.

Project goal:

Numerically simulate shutdown and start-up procedure of the Francis-99 turbine and validate the results through experimental data.

1.3 Limitations

- The work is to be conducted during and limited to the spring term of 2018.
- No physical experiments are to be performed, the project will rely on the ones available.
- All simulations will be carried out using OpenFOAM, i.e. no other method to attain results are to be used.
- The study will only focus on the available Francis-99 design, i.e. no other turbine design will be investigated.
- The computations are to be done on Vattenfall's cluster.
- Turbulence will only be modeled with Large Eddy Simulation (LES).
- No mesh dependency study will be conducted since the mesh has been used previously with reliable results.

1.4 Specification of issue under investigation

- Are LES and OpenFOAM suitable to simulate shutdown and start-up procedures?
- Are the numerical results comparable to experimental results?
- Will LES provide valid results even with a coarse mesh?
- Are the simulations able to capture the expected flow phenomena?
- Is a fast or a slow start-up preferable?
- Can the same be stated for shutdown?
- How is pressure, thrust and torque coupled to the varying mass flow, i.e. guide vane motion speed?

2

Theoretical Framework

2.1 Water Turbines

A water turbine is a machine that converts the potential and kinetic energy of water into a rotating motion. This is then used to drive a generator which produce electricity. The three most common types of hydro turbines are Pelton, Kaplan and Francis. The Pelton wheel is driven by water jets aimed at buckets, basically making it a water wheel. A Kaplan turbine is a propeller shaped design with water flowing past it, while a Francis turbine changes the direction of the flow as it passes the runner [6].

2.1.1 Francis turbine

The Francis turbine is a reaction turbine working in both horizontal as well as vertical direction. Water is led through a spiral casing into a first stage of vanes, see Figure 2.1. These are called stay vanes and have a fixed angle to distribute the water evenly across the full circumference. Next comes the guide vanes, which have the possibility to change angle to guide the water in an optimal direction towards the runner or to shut down the turbine. The runner is the rotating part and thus also the energy extracting part of the turbine. The runner rotates due to the water hitting one side of the runner blades, making it the pressure side, and thus extracting angular momentum and reducing the pressure of the water. The flow direction is turned from radial to axial in the runner, see Figure 2.2 for a side view of the runner. The final stage is the draft tube [6, 7].

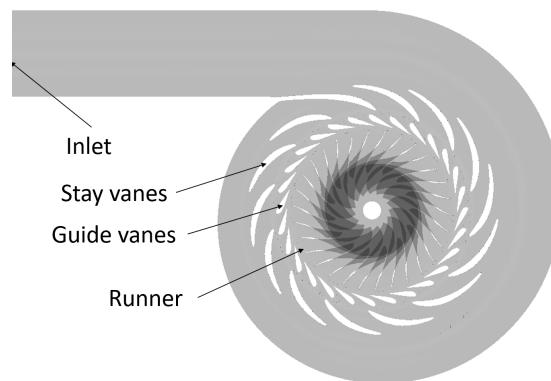


Figure 2.1: Slice of the Francis-99 turbine seen from above.

2.2 Francis-99

To provide academia and developers with a solid platform to do research, the Francis-99 workshops were founded. It is a three stage series of workshops containing both experimental and numerical studies based on the Francis-99 design. The first regards steady state (2014), the second transient conditions (2016) and the final fluid structure interaction, planned to take place during 2018. The turbine is a 1:5.1 scale model of a prototype Francis turbine located at Tokke power plant, Norway [8]. A side view of the turbine is depicted in Figure 2.2. The turbine under investigation has a net head of about $H = 12$ m and a runner diameter $d_{runner} = 0.349$ m. The turbine contains 14 stay vanes, 28 guide vanes and 30 runner blades divided into 15 splitters and 15 full blades [4]. The guide vane angle, α , between minimum load and Best Efficiency Point (BEP) corresponds to 0.80° and 9.84° from completely closed position [3]. Angles between BEP and minimum load are referred to as Part Load (PL). Equation 2.1 demonstrates how the net head is calculated.

$$H_{net} = \frac{p_{1abs} - p_{2abs}}{\rho g} + \frac{v_1^2 - v_2^2}{2g} + z \quad (2.1)$$

where indices 1 represents inlet and 2 outlet, p pressure, v velocity, ρ density, g gravitational constant and z vertical distance between inlet and outlet.

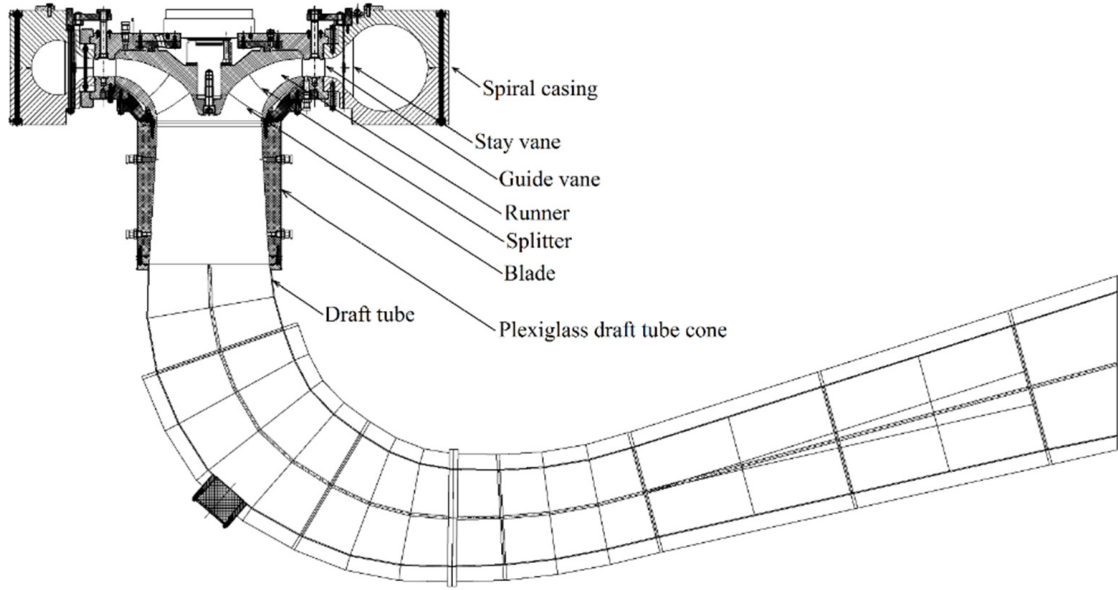
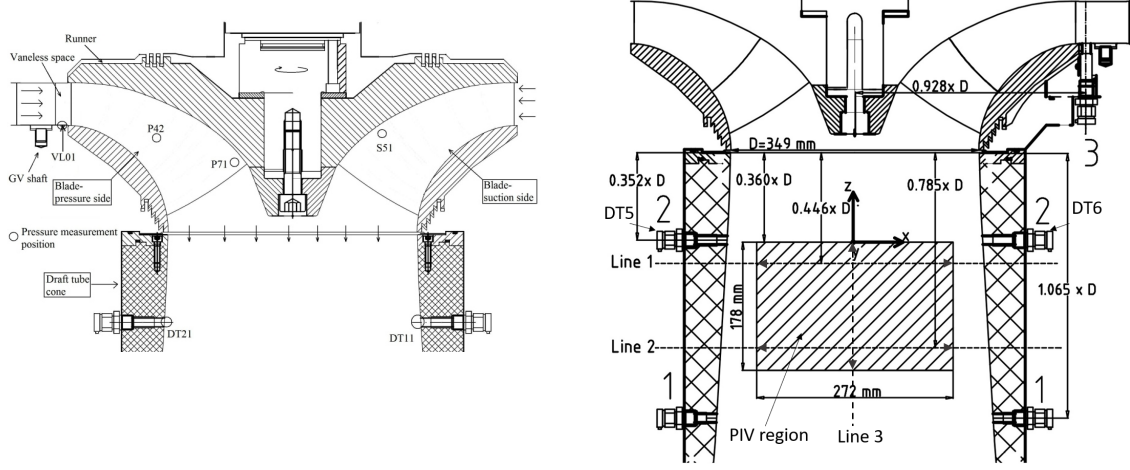


Figure 2.2: Side view of the Francis turbine, source: [8].

2.2.1 Predefined Probes and Velocity Lines

During the Francis-99 workshops a number of measurement locations are determined to compare data between numerical and experimental studies. Figure 2.3 illustrates most of them, Table 2.1 gives the location of all pressure probes and Table 2.2 the velocity lines. The origin of the coordinate system is set so that x and y are zero in

the rotation axis of the runner while z is set to zero at mid height of the runner inlet, also corresponding to mid plane of the guide vanes. Data from the measurement points are available through the Francis-99 Committee [9].



(a) Side view of the Francis-99 with some of the probe locations, source: [8]. (b) Rotated side of the turbine with velocity lines, DT5 and DT6, source: [2].

Figure 2.3: Pressure probe and velocity line locations in two side views. Note that VL2 is not illustrated, but it is placed in a similar manner as VL01, see Table 2.1

Pressure measurements for both shutdown and start up procedure, α at BEP to $\alpha = 0.80^\circ$, are available for probes VL2, DT5 and DT6. The start up and shutdown procedures took approximately 7 s each, the runner had a constant speed of 333 rpm and a net head of about 12 m in both scenarios. Steady BEP measurements for the other probes are accessible from the first workshop where the operating conditions are similar to the ones from the second, but with some minor differences. The location of the pressure probes are found in Figure 2.1 and exact coordinates in Table 2.1, at initial conditions. The runner probes follow the geometrical rotation and are thus not fixed in space.

Table 2.1: Initial pressure probe locations [3, 9]. Note that VL = Vane Less, DT = Draft Tube, P = Pressure side, S = Suction side.

Probe	VL01	DT11	DT21	P42	P71	S51	VL2	DT5	DT6
x mm	262	-90	90	0	-66	-80	-320	-149	149
y mm	194	157	-157	179	42	84	62	-101	101
z mm	-30	-306	-306	-53	-86	-51	-29	-306	-306

Besides the pressure probes, three velocity lines are established in the second workshop. Two horizontal in the draft tube, which are placed at different distances from the runner and one vertical just under the runner, see Figure 2.3b and Table 2.2 for position and coordinates. The lines captured purely axial (vertical) and horizontal (parallel with lines in Figure 2.3b) velocities using Particle Image Velocimetry (PIV) measuring technique. The horizontal velocity is positive in (-0.191; -0.982;

0) direction. The vertical line has 19 measuring points and the two horizontal 28 each. Unfortunately experimental data is only available for the first 18 points at the vertical line, by the Francis-99 committee.

Table 2.2: Position of velocity lines [9].

Velocity line	L1 start	L1 end	L2 start	L2 end	L3 start	L3 end
x mm	25.96	-25.56	25.96	-25.56	0	0
y mm	133.55	-131.49	133.55	-131.49	0	0
z mm	-338.6	-338.6	-458.6	-458.6	-488.6	-308.6

2.3 Fluid mechanics

The field of fluid mechanics describes the motion of flows and how various quantities are transported through and via the fluid. When solving problems in the fluid branch one assumes a continuum and thus derive the governing equations based on a finite volume. By doing this one ends up with the continuity (2.2), momentum (2.3) and energy (2.4) equations, known as Navier-Stokes equations. Via the incompressibility and constant viscosity assumption the equations are written as [10]

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

$$\frac{\partial u_i}{\partial t} + \frac{\partial(u_i v_j)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i \quad (2.3)$$

$$\rho \frac{de}{dt} = -p \frac{\partial u_i}{\partial x_i} + \Phi + \frac{\partial}{\partial x_i} \left(k \frac{\partial T}{\partial x_i} \right) \quad (2.4)$$

where u denotes velocity, t time, x spatial direction, p pressure, ρ density, ν kinematic viscosity, f body force, Φ viscous terms, e internal energy, k kinetic energy and T temperature.

2.4 Computational Fluid Dynamics

Due to the complexity of the governing equations in fluid dynamics, computers are almost exclusively used. The method is commonly referred to as Computational Fluid Dynamics (CFD). The equations are discretized and solved for in a finite volume domain (mesh), see Section 2.4.1. To make the equations less demanding, regarding computational power, various models are applied to the governing equations. The most common is the Reynolds Average Navier-Stokes (RANS), where time average is adopted and all flow structures are modeled. A more accurate method for transient flows are to resolve the largest flow structures and model the smallest, i.e. Large Eddy Simulation (LES). There is also a method called Direct Numerical Simulations (DNS) where all scales and behaviors of the flow are resolved, thus demanding vast computer resources [10, 11].

2.4.1 Finite Volume Method

The numerical technique Finite Volume Method (FVM) is the most commonly used method for computing fluid dynamics. The basics are that one transforms the partial differential equations (PDE), i.e. Navier Stokes equations, to systems of linear equations and integrate them over a finite volume, thus the name. The linearization is achieved via discretization of the PDEs and numerical integration with a certain order of accuracy over the volume [12]. The spatial discretization is conducted through discretization schemes, e.g. up-wind, central difference. For transient simulations, schemes to advance in time are also adopted e.g. Euler implicit, Crank-Nicolson. The system of equations are solved in an iterative manner using e.g. Gauss-Seidel, TDMA [11]. The integral form of the governing equations can be written as (continuity and momentum) [13]

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho \mathbf{u} \cdot \mathbf{n} dS = 0 \quad (2.5)$$

$$\frac{\partial}{\partial t} \int_V \rho \mathbf{u} dV + \int_S \rho \mathbf{u} \mathbf{u} \cdot \mathbf{n} dS = \sum \mathbf{f} \quad (2.6)$$

where ρ denotes density, \mathbf{u} velocities in spatial direction, S surface of finite volume, \mathbf{n} surface normal, V volume of the finite volume and \mathbf{f} surface and body forces acting on the finite volume.

2.4.2 Large Eddy Simulation

As earlier stated, LES is a method to simulate turbulence where the large turbulent scales are resolved while the smaller are being modeled. The modeling is done through a low-pass filter where the small scales are removed by filtering in space. Low-pass in this context meaning that scales associated with high frequencies are filtered out, i.e. small eddies. The filtering can be done in a number of ways. The simplest however, is by using the computational grid. All scales smaller than the element size are automatically filtered since there is no way to capture their behavior between the nodes. This is called box filtering and is the most common filtering technique, i.e. the filter width Δ equals the element size [10].

When filtering, denoted by $\bar{\cdot}$, the momentum equation reads

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial \bar{u}_i}{\partial x_j} \right) \quad (2.7)$$

To make the expression less cumbersome one want to filter only the term, not the gradient of the terms. This may be done with second order of accuracy accordingly

$$\frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = \frac{\partial}{\partial x_i} \left(\frac{1}{V} \int_V u_i u_j dV \right) + \mathcal{O}(\Delta^2) = \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} + \mathcal{O}(\Delta^2) \approx \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} \quad (2.8)$$

Analogous for the other terms. $\overline{u_i u_j}$ is still unknown, this is solved by subtracting $\frac{\partial \overline{u_i u_j}}{\partial x_j} - \frac{\partial \overline{u_i} \overline{u_j}}{\partial x_j}$ on each side of Equation 2.7. The subtracted term represent the Sub-Grid Scale (SGS) stresses and needs to be modeled since the small scales are removed when employing the box filter [10]. Equation 2.7 now reads

$$\frac{\partial \overline{u_i}}{\partial t} + \frac{\partial \overline{u_i u_j}}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial \overline{u_i}}{\partial x_j} \right) - \frac{\partial \tau_{ij}}{\partial x_j} \quad (2.9)$$

where

$$\tau_{ij} = \overline{u_i u_j} - \overline{u_i} \overline{u_j} \quad (2.10)$$

2.4.3 The Smagorinsky Model

The deviatoric part of τ_{ij} is required to be modeled in order to attain closure of the LES equations. The simplest method is to use the Smagorinsky model. In the Smagorinsky model the SGS stress tensor is modeled as follows [10, 14, 15]

$$\tau_{ij} - \frac{\delta_{ij}}{3} \tau_{kk} = -\nu_{sgs} \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) = -2\nu_{sgs} \overline{s}_{ij} \quad (2.11)$$

$$\nu_{sgs} = (C_s \Delta)^2 \sqrt{2 \overline{s}_{ij} \overline{s}_{ij}} = (C_s \Delta)^2 |\overline{s}| \quad (2.12)$$

where δ_{ij} denotes the Kronecker delta, C_s a constant and Δ is typically the filter width i.e. the cell size. The problem with the Smagorinsky model is that C_s is assumed to be constant, but it is in fact flow dependent and thus certainly not constant in a general case [10, 14, 15]. There are however several dynamic models which make C_s dependent of the flow [15].

2.4.4 The Dynamic Model

An alternative to the straight forward Smagorinsky model is to use the dynamic SGS eddy viscosity model where a variable, C , is calculated. A second coarser filter, test filter, denoted by $\widehat{\cdot}$ is introduced. The test filter is defined as twice the original filter width, i.e. $\widehat{\Delta} = 2\Delta$. The momentum equation now has the following form [10, 16]

$$\frac{\partial \widehat{u_i}}{\partial t} + \frac{\partial}{\partial x_j} (\widehat{u_j} \widehat{u_i}) = -\frac{1}{\rho} \frac{\partial \widehat{p}}{\partial x_i} + \nu \frac{\partial^2 \widehat{u_i}}{\partial x_j \partial x_j} - \frac{\partial T_{ij}}{\partial x_j} \quad (2.13)$$

where T_{ij} is the SGS stress with length scales $l < \widehat{\Delta}$, which combined with Equation 2.11 can be written

$$T_{ij} = \widehat{\overline{u_i u_j}} - \widehat{\overline{u_i}} \widehat{\overline{u_j}} = \widehat{\overline{u_i u_j}} - \widehat{\overline{u_i}} \widehat{\overline{u_j}} + \widehat{\tau}_{ij} \quad (2.14)$$

By assuming that T_{ij} can be expressed in the same manner as τ_{ij} in the Smagorinsky model one gets

$$\tau_{ij} - \frac{1}{3} \delta_{ij} \tau_{kk} = -2C \Delta^2 |\overline{s}| \overline{s}_{ij} \quad (2.15)$$

$$T_{ij} - \frac{1}{3} \delta_{ij} T_{kk} = -2C \widehat{\Delta}^2 |\widehat{s}| \widehat{s}_{ij} \quad (2.16)$$

The SGS stresses between Δ and $\widehat{\Delta}$, dynamic Leonard \mathcal{L}_{ij} , is defined as [10, 17]

$$\mathcal{L}_{ij} = \widehat{\overline{u_i u_j}} - \widehat{u_i} \widehat{u_j} = T_{ij} - \widehat{\tau}_{ij} \quad (2.17)$$

C in Equations 2.15 and 2.16 could be compared to C_s^2 from Equation 2.12. By assuming a slow variation of C and applying the test filter to Equation 2.15, one attains

$$\mathcal{L}_{ij} - \frac{1}{3} \delta_{ij} \mathcal{L}_{kk} = -2C \underbrace{\left(\widehat{\Delta^2} |\widehat{s}| \widehat{s}_{ij} - \Delta^2 |\widehat{s}| \widehat{s}_{ij} \right)}_{M_{ij}} \quad (2.18)$$

C is not a constant, but a function of time and space, i.e. $C = C(x_i, t)$. By defining the error, Q , for Equation 2.18 as right-hand minus left-hand and raise it to the power of two

$$Q = \left(\mathcal{L}_{ij} - \frac{1}{3} \delta_{ij} \mathcal{L}_{kk} + 2C M_{ij} \right)^2 \quad (2.19)$$

and differentiate Q with respect to C , thus finding the minimum of the function yields a solution to C

$$\frac{\partial Q}{\partial C} = 4M_{ij} \left(\mathcal{L}_{ij} - \frac{1}{3} \delta_{ij} \mathcal{L}_{kk} + 2C M_{ij} \right) = 0 \quad \Rightarrow \quad C = -\frac{\mathcal{L}_{ij} M_{ij}}{2M_{ij} M_{ij}} \quad (2.20)$$

2.4.5 The One Equation Eddy Viscosity Model

Another alternative to the Smagorinsky model is to introduce an eddy viscosity model based on the kinetic energy, namely the One Equation Eddy Viscosity Model. A transport equation for the kinetic energy, k , is introduced as [18, 19]

$$\frac{\partial k}{\partial t} + \frac{\partial k \bar{u}_i}{\partial x_j} = -\tau_{ij} \bar{s}_{ij} + \frac{\partial}{\partial x_j} \left((\nu + \nu_{sgs}) \frac{\partial k}{\partial x_j} - \frac{c_e k^{3/2}}{\Delta} \right) \quad (2.21)$$

The turbulent eddy viscosity is modeled via

$$\nu_{sgs} = c_k \sqrt{k} \Delta \quad (2.22)$$

SGS stress tensor, τ_{ij} , as

$$\tau_{ij} = \frac{2k}{3} I - 2\nu_{sgs} \bar{s}_{ij} \quad (2.23)$$

where I is a unit tensor. The one equation model is made dynamic in a similar manner as showed in section 2.4.4, yielding [17]

$$c_k = \frac{\mathcal{L}_{ij} M_{ij}}{M_{ij} M_{ij}} \quad (2.24)$$

The other coefficient, c_e , has a constant value of 1.048.

2.5 Mesh Modification

A turbine is closed or opened by turning its guide vanes to decrease or increase the flow rate. This can be simulated making the geometry move on top of an already rotating runner. The guide vanes may be rotated by distorting the mesh as far as possible without getting a mesh quality that generates untrustworthy results.

2.5.1 Laplacian Smoothing

Mesh morphing via a Laplacian Smoothing is governed via the Laplace Equation, Equation 2.25. The motion imposed by the boundary is spread out in the mesh via a diffusivity parameter, γ . Meaning that points close to the moving wall are more effected than those further away [20]. The motion procedure can be decomposed in a number of steps [21]. Firstly, all polyhedral cells are divided into tetrahedral, by splitting faces to triangles, and a cell center point is introduced. The second step is to use the Laplace operator and move points, Equations 2.25 - 2.26.

$$\nabla \bullet (\gamma \nabla \mathbf{u}) = 0 \quad (2.25)$$

Where γ is the diffusivity factor, which decrease with distance from the motion. The diffusivity may be governed linearly, exponentially, inverse distance to the object or via any arbitrary decreasing function. \mathbf{u} is the velocity field of the points, used to modify points position, \mathbf{x} , as

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \mathbf{u}\Delta t \quad (2.26)$$

The next step is to perform the motion of the wall by executing the equation of motion prescribed by the boundary condition. The wall displacement, and thus point displacement, is solved in an iterative manner until convergence is reached [21].

2.6 OpenFOAM

Open source Field Operation And Manipulation (OpenFOAM) is a toolbox built in C++ and is primarily meant to solve continuum mechanics problems. It is however most often used to solve problems in CFD. The source code is fully accessible by the user making it a very powerful and modifiable product, but since there is no interface and no complete documentation the learning curve might be considered steep. OpenFOAM provide all necessary tools for meshing, pre-processing, solving and post-processing [22].

3

Methodology

This chapter presents the methodology used for this study. Pre-processing is described first, followed by post-processing features.

3.1 Simulation Procedure and Work Flow

All simulations were preformed using the open source software OpenFOAM 2.3.x. To begin the transient simulations a well developed and trustworthy initial solution was required. The starting point was the Best Efficiency Point (BEP) since it was possible to compare results from both experimental data and previous studies. The operating point was simulated, with LES, until steady condition was reached. Then the guide vane boundary conditions were changed, via the utility `changeDictionary`, to start closing the guide vanes. The entire shutdown transient was simulated, BEP to the guide vanes being 0.80° from totally closed. When 0.80° was reached the guide vanes were brought to a standstill to see if the "stop closing" had any effect and to attain steady condition again. The new steady solution was a good initial solution for the start-up scenarios. Four different start-up simulations were run with different angular velocity. The start-ups covered a range of approximately 1.90° , from 0.80° to 2.70° .

3.2 Pre-Processing

An accurate and efficient Pre-Processing setup was required to get the simulations running and to generate plausible results. In this section the used models, boundary conditions and procedures are introduced and described. A lot of the settings were chosen based on previous work conducted by Guðjón Vesteinsson in 2015. His simulations showed promising results.

3.2.1 Turbulence Modeling

As mentioned LES was used, see Section 2.4.2 for theory, together with the dynamic one equation eddy viscosity model, described in Sections 2.4.4 and 2.4.5. In OpenFOAM 2.3.x the SGS model is named `dynOneEqEddy`. The cube root volume method, Equation 3.1, was used to determine the filter width Δ .

$$\Delta = V_{cell}^{\frac{1}{3}} \quad (3.1)$$

V_{cell} is the volume of a computational cell.

3.2.2 Solver Settings

OpenFOAM allows the user to apply different types of solvers for different entities. This project utilized the same solvers as the thesis work as it was based on, Guðjón Vésteinsson 2015, see Table 3.1.

Table 3.1: Solvers used for different entities.

	Solver	Smoother
Pressure, p	GAMG	DICGaussSeidel
Velocity, u	smoothSolver	symGaussSeidel
Kinetic energy, k	smoothSolver	symGaussSeidel
Cell displacement	PCG	DIC

GAMG is an abbreviation of Geometric Agglomerated Algebraic Multigrid, PCG for Preconditioned Conjugate Gradient and DIC for Diagonal-based Incomplete Cholesky. All of these are different ways of iteratively solving a system of equations [24].

3.2.3 Pressure and Velocity Coupling

PISO and SIMPLE loops were used to establish a link between pressure and velocity. Combining both yields a PIMPLE loop, where PISO is used as an inner loop and SIMPLE as an outer, i.e. a SIMPLE loop was performed on every PISO loop [29]. PIMPLE was used extensively, only PISO was tested but generated very noisy data in comparison to the more complex ditto. PIMPLE was set to operate with two `nCorrectors` and maximum of ten `nOuterCorrectors`. Meaning two PISO loops within every SIMPLE loop and a maximum of ten SIMPLE loops. Residual controls governed how many SIMPLE loops were required in the time step, i.e. if PIMPLE had reached a specified tolerance it would break and advance the simulation in time without performing the remaining SIMPLE loops. For p tolerance = 1E-5 and relative tolerance = 0, for U tolerance = 1E-6 and relative tolerance = 0.1.

3.2.4 Numerical Schemes and Time Step

Different schemes were used for time and space and during simulation start and running simulation, see Table 3.2. The reason was that a 2nd order scheme in time and divergence of velocity were too accurate to start from. Hence 1st order was used for the first iterations as it was more diffusive. Later the two schemes were changed to 2nd order accuracy. In time Crank Nicholson, with an attribute value of 0.5 was used. This making it more stable via combining pure Crank Nicholson and the Euler forward schemes [25].

Table 3.2: Numerical schemes used for the simulations. Note that schemes in parentheses, e.g. (Euler), refers to the scheme used the first iterations.

ddtSchemes	
default	CrankNicolson 0.5, (Euler)
gradSchemes	
default	Gauss linear
grad(p)	Gauss linear
grad(U)	cellLimited Gauss linear 1
divSchemes	
default	none
div(phi,U)	Gauss linearUpwind grad(U), (Gauss upwind)
div(phi,k)	Gauss upwind
div((nuEff*dev(T(grad(U)))))	Gauss linear
laplacianSchemes	
default	Gauss linear limited corrected 0.33
interpolationSchemes	
default	linear
snGradSchemes	
default	limited corrected 0.33

The time step was controlled by an adjustable utility, `adjustTimeStep`, which make sure that the Courant number, Co , was not too high. Co is a non-dimensionless quantity measuring how far information has traveled per time step, see Equation 3.2. When starting a simulation a max Co of 1 was used. But as the simulation progressed it was increased to a maximum Co of 30 - 50. This resulted in a time step of about 2.5E-4 - 3.5E-4 s and a mean Co in the order of 1E-2.

$$Co = \frac{u\Delta t}{\Delta x} \quad (3.2)$$

3.2.5 Boundary Conditions

To establish good results it was important to set up proper Boundary Conditions (BC). Hints could be found in previous studies of turbines, with and without moving guide vanes [4, 26, 27]. All vital BCs used can be found in Table 3.3. Since water turbines commonly operate due to pressure difference between inlet and outlet, this was to be the case also in this study. Meaning that mass flow were allowed to vary with the operation point, just as in the experimental case. A total pressure boundary condition was applied to both inlet and outlet. This enabled the static pressure to vary slightly due to decreasing/increasing volume flow rate, i.e. velocity of the fluid. The inlet total pressure was tuned so that the volume flow rate of the simulation mimics the one of the experimental as close as possible. The outlet total pressure was based on absolute pressure and velocity from the experimental test case according to Equation 3.3.

$$p_0 = p_{abs} + \rho \frac{1}{2} \left(\frac{\dot{V}}{A} \right)^2 \quad (3.3)$$

where $p_{abs} = 110$ kPa was chosen, $\dot{V} = 0.19959$ m³/s and $A = 0.236$ m². The other boundaries were prescribed by a zero gradient pressure. Velocity utilized a no-slip condition everywhere except for inlet and outlet. There a pressure inlet velocity condition was used, essentially meaning zero gradient except for the initial solution. The kinetic energy, k , and turbulent eddy viscosity, ν_{sgs} , was set in accordance to the numerical work done by Guðjón Vésteinsson in 2015. These correspond to a small value at the walls for k and zero gradient for ν_{sgs} [27]. See Table 3.3 for inlet and outlet BCs of the various entities. A constant density and kinematic viscosity was used with the subscribed values $\rho = 999.8$ kg/m³ and $\nu = 9.57\text{E-}7$ m²/s.

Table 3.3: Inlet and outlet boundary conditions. The u_x values are for BEP. Note that OpenFOAM use a kinematic pressure, $p^* = p/\rho$, for incompressible solvers.

	Inlet	Outlet	Unit
p_0^*	totalPressure, 232.5	totalPressure, 110.8	m ² /s ²
u	pressureInletVelocity, $u_x=2$	pressureInletVelocity, $u_x=1$	m/s
k	fixedValue, uniform 0.01	zeroGradient	m ² /s ²
ν_{sgs}	zeroGradient	zeroGradient	m ² /s

3.2.6 Transient Conditions

The runner had a prescribed rotational speed of 333 RPM for the simulations, see Table 3.4 for all transient operation conditions. For the shutdown case the guide vanes were closed with an angular velocity of 1.3°/s to match the available experimental data. The start up case utilized four different guide vane opening speeds: 0.965°/s, 1.30°/s, 1.90°/s and 2.60°/s.

Table 3.4: Transient operation conditions.

Parameter	Value	Unit
Runner speed	333	RPM
Shutdown GV angular velocity	1.30	deg/s
Start up case 1 GV angular velocity	0.965	deg/s
Start up case 2 GV angular velocity	1.30	deg/s
Start up case 3 GV angular velocity	1.90	deg/s
Start up case 4 GV angular velocity	2.60	deg/s

3.2.7 Mesh

The mesh used was originally created by Guðjón Vésteinsson during his master's thesis in 2015. It is a relatively coarse mesh but it has been used to perform similar calculations and done so providing trustworthy results. It was designed using the

OpenFOAM extension `cfMesh`, an automated mesh generator. The geometry was divided into four parts: Spiral Casing and Stay Vanes (SC-SV), Guide Vanes (GV), runner, and Draft Tube (DT). The parts were defined as two rigid parts (SC-SV and DT), one rotating (runner) and one distorting part (GV). The cell motion imposed by the guide vanes were allowed to spread out in SC-SV part of the mesh, but SC-SV did not host any mesh motion on its own. The parts were joined using the OpenFOAM utility `mergeMeshes` and by prescribing a `cyclicAMI` BC between parts. AMI connect values from a mesh domain to an adjacent by calculating a weighted average from the overlapping faces, thus continuity is fulfilled. Errors and bad overlapping will exist in the model but the overall error is considered small [23]. The total mesh contained approximately 20 million cells, see Table 3.5 for number of million cells in the each part and for different GV meshes.

Table 3.5: Number of million cells per mesh part, total number of cells for the entire mesh was about 20 million. Note that the guide vane angle is based on degrees from closed position.

Part	SC-SV	Runner	DT	GV BEP	GV 4.67°	GV 2.00°	GV 0.80°
10E6 cells	4.60	9.36	1.09	5.70	5.63	5.51	4.67

Since the turbine was simulated during varying load cases the guide vane angle required to be modifiable. This was why that part of the mesh was distorted. As long as the mesh quality was tolerable, a single distorting mesh was used. But at a certain point a new one needed to be generated, see Section 3.2.7.2. When the switch between guide vane meshes was performed the rotation of the runner also required to be taken into consideration. Finally the solution had to be mapped from the previous, see Section 3.2.7.3. To make the mesh generation smooth three `bash` scrips were developed: one which managed the creation of the guide vane meshes, Appendix A.1, a second to merge the different parts and rotate the runner, Appendix A.2, and a final to set up the case, Appendix A.3.

3.2.7.1 Mesh Motion

Both runner and guide vanes utilized mesh motion but in different ways. The guide vane displacement was controlled by a boundary condition while the runner rotation was set by the solver itself. To allow for runner motion the OpenFOAM solver `pimpleDyMFoam` was modified by Eric Lillberg. The rotation was enabled via making all points in the runner frozen and performing a solid body rotation about its rotational axis, see Appendix A.4. The guide vane motion was controlled by a modified point displacement boundary condition. This was basically done by removing a sine term in the `angularOscillatingVelocity` BC, thus making it non-oscillating, see Appendix A.5. The guide vanes used a displacement laplacian, read Section 2.5.1 for info, to morph the mesh. The mesh motion was spread out in the mesh via the `inverseDistance` diffusivity.

3.2.7.2 Generate Guide Vane Mesh

Since it was unknown how far one could rotate the guide vanes before the mesh quality was inadequate, a smooth and preferably automated mesh generation procedure was desirable. Hence the creation of a number of `bash` scripts which could manage this task, see Appendix A.1 for guide vane mesh script. Surface files (.stl files) for PL (3.91°) and center of rotation for the guide vanes were available through NTNU and the previous master's thesis. The position of the main guide vane, `gv_01`, was transformed from its local origin to the global and rotated around z -axis to the desired position. Later `gv_01` was translated back to its original position and copied in an anticlockwise direction to create the other guide vanes and match each center of rotation. All guide vanes were collected to a single .stl file during the copying. When the guide vanes had been merged into a single file the remaining parts (GV-inlet, -outlet, -upper and -lower) were added to a complete guide vane .stl file. Then modified `cfMesh` files were used to generate the guide vane meshes for various guide vane angles.

It was not possible to generate a good mesh with a reasonable amount of cells for the lowest guide vane angle, 0.80°, with `cfMesh`. Instead the automatic OpenFOAM mesher `snappyHexMesh` was used to generate an adequate mesh for that angle. This mesh was provided and created by Eric Lillberg.

3.2.7.3 Replace Guide Vane Mesh and Map Fields

To transfer a field of values from one simulation to another OpenFOAM has the `mapFields` utility. The tool interpolate values from a source mesh to the closest corresponding nodes in a target mesh. When switching between guide vane meshes this was used to continue the simulation. By doing this one had to take the rotation of the runner into consideration due to the fact that its points were rotated and did not have the same origin as the initial case. Therefore, the runner of the target mesh was modified to match the rotation of the source mesh via the `transformPoints` utility. The pressure probe positions of those connected to the runner (P42, P71 and S51) were also updated via a simple rotation matrix according to Equation 3.4. See Section 2.2.1 for definition of probes and locations.

$$\mathbf{P}_{new} = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \mathbf{P}_{old} \quad (3.4)$$

where \mathbf{P} is the x, y, z coordinates of the probe and γ the new rotation of the runner.

When changing guide vane mesh it was essential to use initialization values for k and p similar to the source, otherwise the mapped solution would not work. This was due to that if a target node was outside of the source domain the initialization value was prescribed to it. This happened to some nodes at the `cyclicAMI` surfaces connecting the guide vanes to the rest of the mesh.

To minimize the number of meshes required, the guide vane angle in the target mesh was set so that it matched the last angle in the source mesh. This meant that the angle in the target mesh had to be increased before mapping the solution. Figure 3.1 depicts an illustrative view of this procedure and Table 3.6 shows at which angles a new mesh was introduced. In the figure α is the angle which the source mesh was able to rotate the guide vanes to before the mesh quality was too low. β is the angle one needed to increase the target mesh with to match the source. This pre-rotation, before using `mapFields`, was implemented by taking one time step using `pimpleDyMFoam`. The time step was in the order of $dt \approx 1E-10$ s. It was tried with `moveDynamicMesh`, but then the initialization values were not transferred to the new time step. Thus generating a bad mapping since some cells were not able to be mapped straight away.

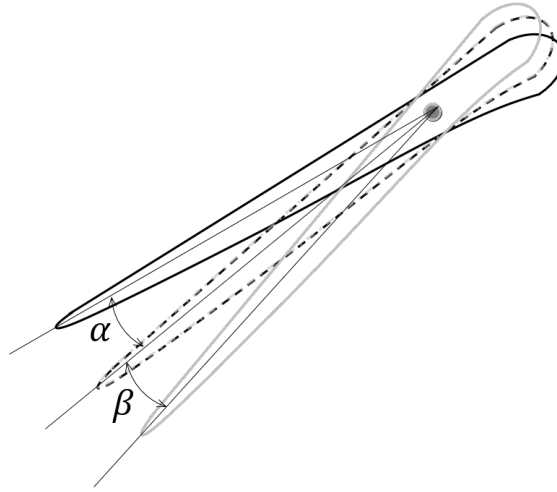


Figure 3.1: Illustrative view of matching guide vane angles between meshes, black is source and gray target mesh.

Table 3.6: Angle simulation span for the guide vane meshes, value in parentheses represents the start-up cases.

GV Mesh	BEP	4.67°	2.00°	0.80°
Span	9.84° - 6.67°	6.67° - 3.68°	3.68° - 1.47°	(2.70°) 1.47° - 0.80°

When starting the simulation after a mapping was complete one had to change schemes in time and divergence of velocity to first order accurate. This to ensure that the solution did not crash. Later the schemes were changed to second order. The time step was at the beginning required to be low, in the magnitude of about $1E-8$ s and slowly ramped up in a succession of stages. The procedure was conducted through first prescribing a fixed time step and then change to the adjustable with a max Co of 1. When the time step had stabilized the Co was further increased to 30 - 50 depending on what the residuals indicated.

3.2.7.4 `flattenMesh`

In a Francis turbine the guide vanes only rotate in a single plane, the xy -plane in this geometry. Therefore, the z -coordinate of each node should and could be constant while rotating. The mesh tool was not accurate enough, all surface nodes were not at the same z -coordinate, for the guide vanes upper and lower parts. This caused problems since the solver then tried to morph cells in the z -direction as well, generating not only longer computational time but also a bad mesh morphing. To force all nodes to have the same value at the upper and lower guide vane surfaces a modified `flattenMesh` was used, see Appendix A.3.2 for the modified utility.

3.3 Post-Processing

An efficient post processing sequence was necessary to analyze and present the attained data in a sufficient manner. In this section the various utilities used are presented. All post processing was performed using Paraview or Matlab.

3.3.1 Probes

The same spatial locations for pressure probes and velocity lines was utilized as for the experiments conducted by NTNU, see Section 2.2.1. Note that all probes were fixed except those connected to the runner. Values from the experimental study was acquired with a sampling frequency of 5 kHz for pressure and 40 Hz for velocity. This while the numerical study uses a varying time step at approximately $3\text{E-}4$ s, corresponding to a frequency of 3.33 kHz. For the steady case mean values were compared while the transients compared time dependant data. After the time scale in simulation and experiment was matched it was rather straight forward to compare the experimental and simulated results. This except the horizontal velocity component, which was positive in $(-0.191; -0.982; 0)$ direction, i.e. not entirely in x or y . To solve this both x and y velocity components were used along with a rotation matrix.

3.3.2 Fields

Probes register a single value at a certain location in space and time. It was hard to draw conclusions from probe data alone, which was why field plots of velocity and pressure were also used. This since they better depicted what happened and how the fluid behaved in the entire domain. They were also analyzed to ensure that the flow had reached steady conditions before commencing the transient. Fancy snapshots were generated to show the most significant flow features.

3.3.3 Q-Criterion

OpenFOAM was able to calculate the vorticity of the fluid via the Q -criterion. This was later represented in the form of an iso-surface to get a grip of flow structures in the domain. Q was set to an iso-value of 200 and is defined as [28]

$$Q = \frac{1}{2} ||\Omega_{ij}||^2 - ||s_{ij}||^2 \quad (3.5)$$

where Ω represent the vorticity tensor and s the rate-of-strain tensor [10].

3.3.4 Mean and Fluctuating Values and Standard Deviation

An absolute quantity, x , can be decomposed into a mean and a fluctuating value accordingly [10]

$$x = x_{mean} + x_{fluct} \quad \Rightarrow \quad x_{fluct} = x - x_{mean} \quad (3.6)$$

where the mean value is calculated as

$$x_{mean} = \frac{1}{N} \sum_{i=1}^N x_i \quad (3.7)$$

where N is the number of samples.

The Standard Deviation, SD, is calculated via

$$SD = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - x_{mean})^2 \right)^{1/2} \quad (3.8)$$

3.3.5 Forces

Also used was a `functionObject` called `forces`. It was added to the code via `controlDict` in the same manner as the probes. With the difference that this utility calculate forces and moments associated to turbo machines instead of just extracting values. The forces acting on the runner were defined as the sum of all pressure forces, F_p , and viscous forces, F_v , acting on the faces connected to the runner surface. The same applies to the pressure moments, M_p , and viscous moments, M_v [30]. These are calculated as

$$F_p = \rho \sum p \mathbf{A} \quad (3.9)$$

$$F_v = \sum \mathbf{A} \nu \quad (3.10)$$

$$M_p = \rho \sum \mathbf{V} \times p \mathbf{A} \quad (3.11)$$

$$M_v = \sum \mathbf{V} \times \nu \quad (3.12)$$

where \mathbf{A} is the area normal vector of each face on the runner, ν effective viscous stress tensor and \mathbf{V} location of the face. Vectors are defined with respect to the origin of the geometry, i.e. the center of the runner. The total force, F , and moments, M , subjected to the runner was the sum of pressure and viscous terms according to Equations 3.13 - 3.14.

$$F = F_p + F_v \quad (3.13)$$

$$M = M_p + M_v \quad (3.14)$$

3.3.6 Data Filter

Most of the data presented in Chapter 4 are raw unfiltered data. But when a data series contained too much white noise a smoothing filter was applied. This was true for all horizontal velocity components, the velocity contour plots and pressure probe VL2 for the final 0.80° guide vane mesh. The filtering was done using the `smooth` function in Matlab. It utilizes a moving average filter and a filter width of 3 - 20 depending on how noisy the data was. The `smooth` function filters data according to [31] and an example with a filter width of 5 is shown in Equation 3.15.

$$x_{smooth}(i) = \frac{1}{N} \sum_{n=-2}^{N=5} x(i+n) \quad (3.15)$$

Where x is an arbitrary function/data and N the filter width.

3.3.7 Fast Fourier Transform

The Fast Fourier Transform (FFT) can be used to convert any time sampled entity into the frequency plane. This to analyze whether certain fluctuations keep reappearing forming a pattern. Matlab was used to perform this conversion and it uses the following algorithm [32]

$$Y(k) = \sum_{j=1}^N X(j) W_N^{(j-1)(k-1)} \quad (3.16)$$

where $X(j)$ is the original signal and W_N is defined as

$$W_N = e^{(-2\pi i)/N} \quad (3.17)$$

FFT required an equidistant time step, i.e. a constant sample frequency, to produce any possible results. But during the transients an adjustable time step was used, meaning that the sample frequency varied. To overcome this problem the Matlab function `resample` was used. It interpolates data with a non-uniform frequency to a desired uniform, using an anti-aliasing filter [33].

4

Results and Discussion

Results are presented for the initial steady case, guide vane meshes and mesh morphing, shutdown, and start-up cases. Each of the simulation sections, i.e. not guide vane meshes and mesh morphing, is concluded in a final part where the results are further discussed and evaluated.

4.1 Steady at Best Efficiency Point

The results presented in this section are for the "steady" run at BEP, steady in the sense that the guide vanes are kept stationary. The runner has a constant rotational speed of 333 rpm. The volumetric flow rate is tuned to match the one used in the experimental case. Flow rates at BEP are $\dot{V}_{BEP,SIM} = 0.1965 \text{ m}^3/\text{s}$ and $\dot{V}_{BEP,EXP} = 0.1996 \text{ m}^3/\text{s}$, yielding a difference of 1.55 %. This is deemed as accurate enough since it is a tedious process to tune the flow rate via the total pressure boundary condition. The \dot{V} difference may be a source of error when comparing results.

4.1.1 Pressure Probes

A comparison between mean static pressure for numerically simulated and experimentally measured data is depicted for several probes in Figure 4.1. The results are in general quite accurate, but a small deviation is clearly noted for all probes except VL2. It could partly be explained by that the operating conditions between the experiment and simulation are slightly different for all probes except that one. This since the measurements for the other probes are from the first Francis-99 workshop and VL2 from the second. The simulation shows similar behaviour but it demonstrates higher values, especially for VL01, S51 and P71.

The fluctuating pressure attained in DT5 and DT6 coincides to a satisfying extent for simulation and experiment. No static pressure is available from the experiment for DT5 and 6, but the operating conditions are identical regarding runner speed and material properties. The Standard Deviations (SD) at DT5 are 0.251 and 0.242 for simulation and experiment respectively. This yields a difference of only 3.72 %. The corresponding for DT6 is 3.71 %. Figure 4.2 shows the pressure fluctuation and SD for a short time, it can be noted that the peak fluctuations are in the same order of magnitude for simulation and experiment.

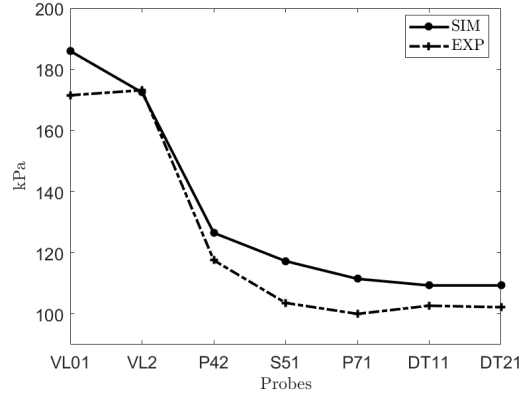


Figure 4.1: Mean static pressure, numerically simulated (SIM) and experimental (EXP), data in various pressure probes.

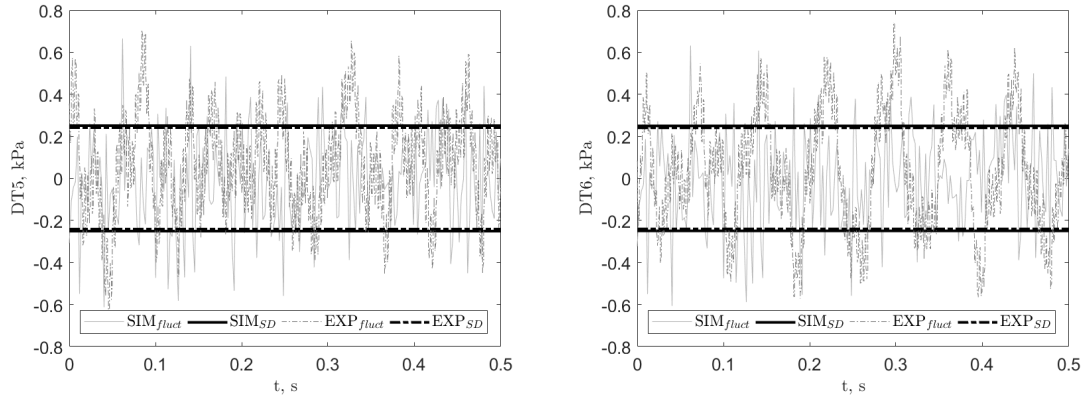


Figure 4.2: Pressure fluctuations and SD for DT5 and DT6 over time.

4.1.2 Velocity Probes

LES-generated axial (vertical) and horizontal velocities operates in the same order of magnitude as the experimental ditto, compared in Figure 4.3. The shapes and values of the axial velocity coincide rather well, except in the center of the draft tube. The decrease of axial velocity close to the rotational axis is explained by the runner having a flat spot at the center causing a wake. This is where the runner is attached to the shaft. It seems that the simulation over predicts this decrease in relation to experiment. It should be noted that the experiment could under predict the decrease since the uncertainty of the measurements may increase in regions of low velocity. The symmetric shape of the axial velocity is also nicely captured.

The horizontal components deviate a bit more in shape from each other. The characteristics of the simulated counterpart seems highly reasonable due to that it is almost perfectly anti-symmetric around the center, see Figure 4.3. This is an expected behaviour since the runner introduce some centrifugal force to the fluid. Thus making it go in radial direction, away from the centre. The flatter shape of the experimental data indicates that there is less radial velocity present than in the simulated case.

Meaning that more flow leaves the turbine axially in the experiment. This could be a contribution to the small differences in the axial velocity.

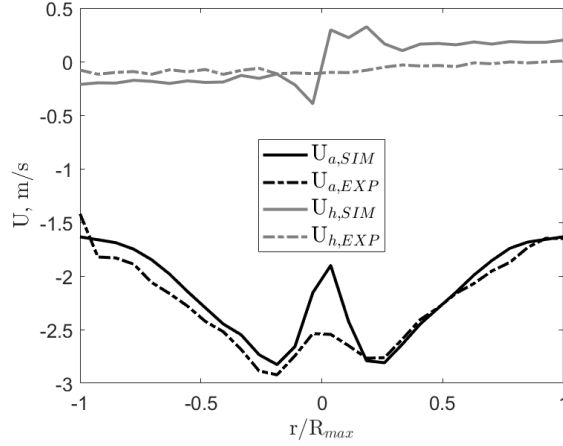


Figure 4.3: Mean velocity, simulated and experimental, at velocity line 1. a = axial velocity and h = horizontal velocity, R_{max} is the largest radial position at the velocity line, not DT wall, see Figure 2.3b.

4.1.3 Velocity Field

The velocity distribution and general behaviour of the fluid is as depicted in Figure 4.4. Fairly constant in the spiral casing and over the stay vanes, increasing over the guide vanes to the runner, dropping in the runner and slowed down to its final velocity in the draft tube. Worth pointing out is the rather large velocity gradient in the bend of the draft tube. It is as if the fluid does not really make use of the whole cross section of the tube and there are areas with a velocity magnitude close to zero. Also worth noting is that the bend of the draft tube is too steep for the fluid and a behaviour comparable to separation is therefore induced. This generate fluctuations downstream. If one zoom in to the region beneath the runner it is actually possible to see part of the wake discussed in Section 4.1.2. In this particular snapshot in time it is quite hard to see.

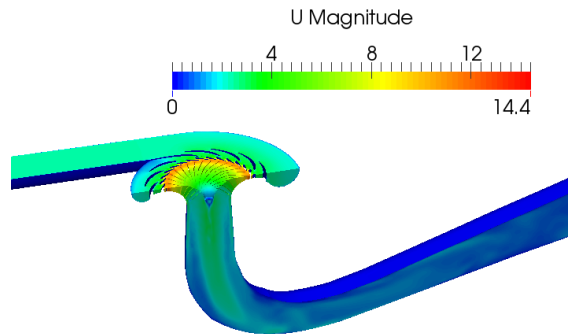


Figure 4.4: Cut view of the velocity magnitude in the domain.

4.1.4 Static Pressure Field

Most of the static pressure is lost when the fluid accelerates through the guide vanes and passing through the runner. The conversion between high static pressure and low velocity and vice versa seems reasonable, compare with velocity in Figure 4.4. The distribution of the static pressure loss is evenly divided over the guide vanes and runner. Figure 4.5 illustrate a cut view of the domain where these feature can be seen.

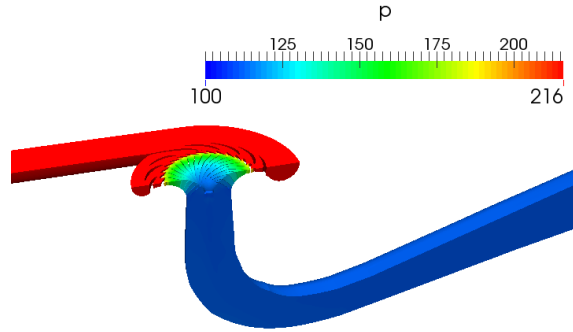


Figure 4.5: Cut view of the kinematic static pressure field in the domain.

4.1.5 Comments About the Steady Results

A steady simulation is required to have reliable results to initialize the following transients. Generally the results are rather good and comparable to the experimental data. Worth noting is that this study utilize pressure measurements from two different experimental studies, thus perhaps explaining some differences in results. OpenFOAM display higher overall losses since the inlet pressure boundary condition had to be increased to a larger value than the experiments to match the volumetric flow in the turbine. The experimental case utilized an inlet static pressure of about 216 kPa [2]. Even though some effort was put in to the matching of flow rates, slight differences still exist. This is without a doubt one reason to why there are some deviation in the results.

There is a difference in axial velocity at the center of the draft tube, as seen in Figure 4.3. This might be due to some inaccuracy in the experimental test case or that the numerical simulation is inadequate in that region. It is however reasonable to have a velocity drop in the center. This since the wake below the runner affects the flow in regions where the velocity lines are. Beside this minor doubt, the results are deemed accurate enough to commence the transients.

4.2 Guide Vane Meshes and Mesh Morphing

The guide vane meshes used for the transient simulations are introduced in this section. First, the BEP guide vane mesh is visualized along with example snapshots of the morphed BEP mesh, see Figure 4.6. Then the other used guide vane meshes,

4.67°, 2.00° and 0.80°, are showed in an unmorphed state, see Figure 4.7.

The guide vane mesh used for BEP can be rotated approximately 3.0° before the mesh quality is inadequate. Figure 4.6 shows the BEP guide vane mesh at 0.00°, 1.50° and 3.00° rotation from its original position. These correspond to 9.84° (BEP), 8.34° and 7.24° from closed position, i.e. α .

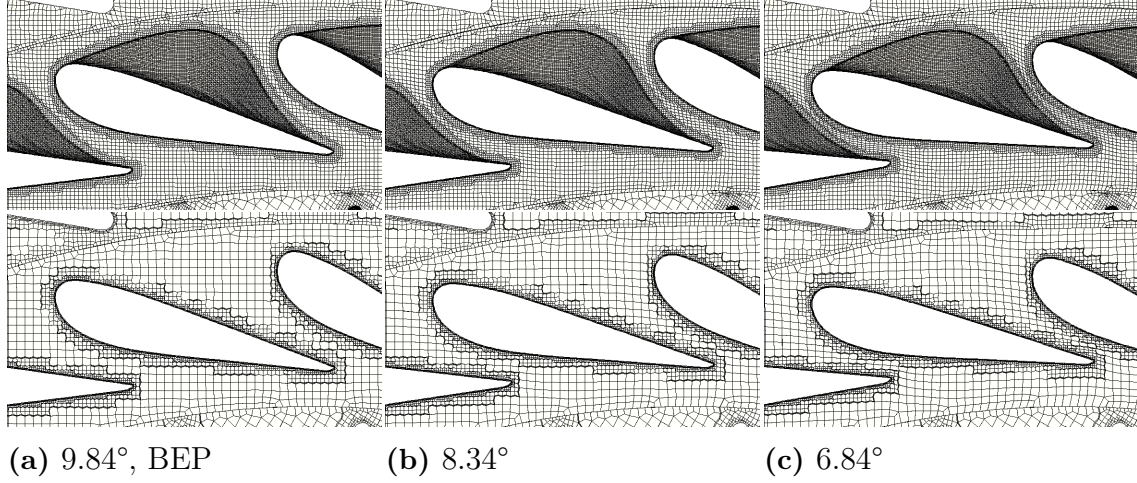


Figure 4.6: Morphed BEP guide vane mesh, above top view and below mid plane. Note that the angles corresponds to α , i.e. degrees from completely closed position.

Guide vane meshes at 4.67° and 2.00° are similar to the BEP mesh, compare Figures 4.6, 4.7a, and 4.7b. This since they are generated using the same strategy via `cfMesh`. The final 0.80° guide vane mesh, Figure 4.7c, shows some differences in ratio between adjacent cells compared to the others. This due to that it is generated using another strategy in `snappyHexMesh`. The mesh difference may explain why e.g. the pressure probe data in VL2 had to be filtered to match data from previous simulations, discussed later in Section 4.3.

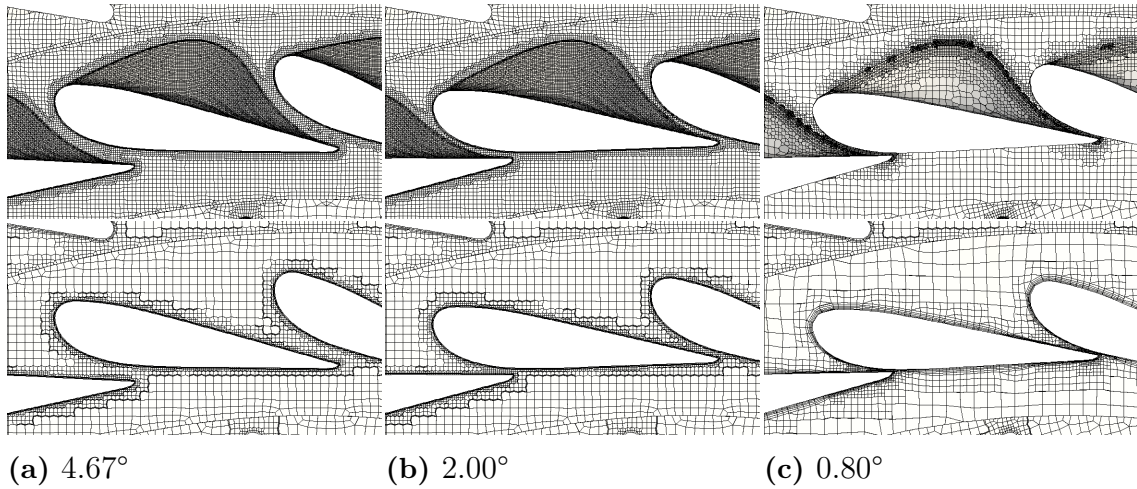


Figure 4.7: Unmorphed guide vane meshes used in the simulations. Note that the angles corresponds to α , i.e. degrees from completely closed position.

4.3 Shutdown Transient

In this section the transient data for the shutdown case is presented and discussed. It should be noted that the simulated time has been manipulated to match the "start to close time" from the experiment. In the experimental test case the guide vanes are starting to close after one second. Figure 4.8 is a view of the simulated and experimental guide vane angle, α , during shutdown. They do not match perfectly since the simulation use an immediate and constant change of guide vane angle while the experimental rig is accelerated to the opening speed. The figure can also be used for conversion between time and α since both are used in this section.

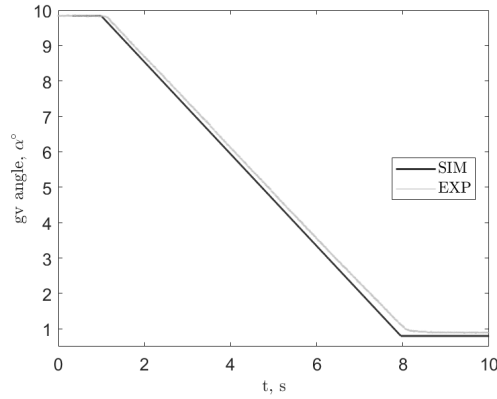


Figure 4.8: Guide vane angle change during shutdown in experiment and simulation.

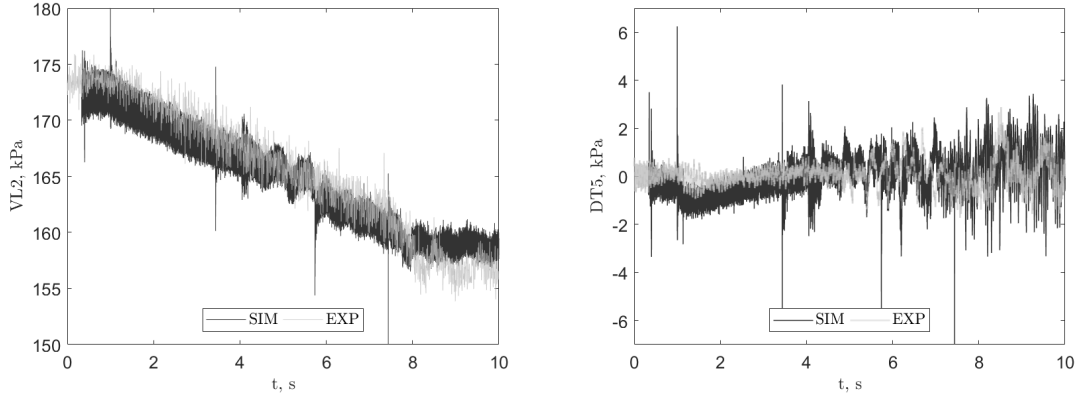
The volumetric flow rates are $0.1996 \text{ m}^3/\text{s}$ and $0.1965 \text{ m}^3/\text{s}$ for experiment and simulation before the closing commence, a difference of 1.55 %. The rates after shutdown are $0.0220 \text{ m}^3/\text{s}$ and $0.0168 \text{ m}^3/\text{s}$, which corresponds to a difference of 23.6 %, i.e. the deviation has increased but is still in the right region. Hence a verification of that the total pressure boundary condition worked as intended. It is reported that air bubbles were present at lower guide vane angles in the experimental case [2]. The uncertainty of the measurements are thus higher at lower α .

4.3.1 Pressure Probes

Transient data for pressure probes VL2 and DT5 are displayed in Figure 4.9 during the shutdown sequence. Generally the simulated data matches the experimental to a great extent. But as can be seen clearly in Figure 4.9a there is a small difference. The pressure is slightly under predicted by OpenFOAM during the shutdown sequence. This could possibly be explained by that the volumetric flow rate is not exactly matched between simulation and experiment. The mean values for VL2, before closing, are 172.4 respectively 173.2 kPa for simulation and experiment, i.e. a difference of only 0.8 kPa, corresponding to a difference of 0.4 %. If one adds this difference to the simulated transient value both simulation and experiment matches almost identically in the mean value. When the closing sequence is complete, at $t \approx 8 \text{ s}$, the simulation change behavior and over predict the pressure at VL2 in

comparison the experimental data. This may partly be explained by that another type of guide vane mesh is used for the final angle change and steady part, $t > 7.3$ s. The spikes present in Figure 4.9a corresponds to a change of guide vane mesh. A pulsating trend is also captured at $t > 4$ s.

The fluctuating pressure in the draft tube, Figure 4.9b, occur in a similar fashion between simulation and experiment. But the magnitude of the fluctuations seems to be somewhat over predicted by OpenFOAM. One can also see where the simulation has been stopped/started and where new sets of guide vane meshes have been introduced due to the spikes and since it takes some time before the pressure to stabilize. The fluctuations gets worse as the guide vane angle decrease below roughly 3.5° , $t \approx 6$ s. Also seen is that the experimental case show a cyclic appearance at the lower angles. Both the larger fluctuations and lack of a clear cyclic behavior in the simulation is probably due to that the flow is more turbulent in nature at small angles or that the mesh in the draft tube is too coarse.



(a) Static pressure for probe VL2.

(b) Fluctuating pressure for probe DT5.

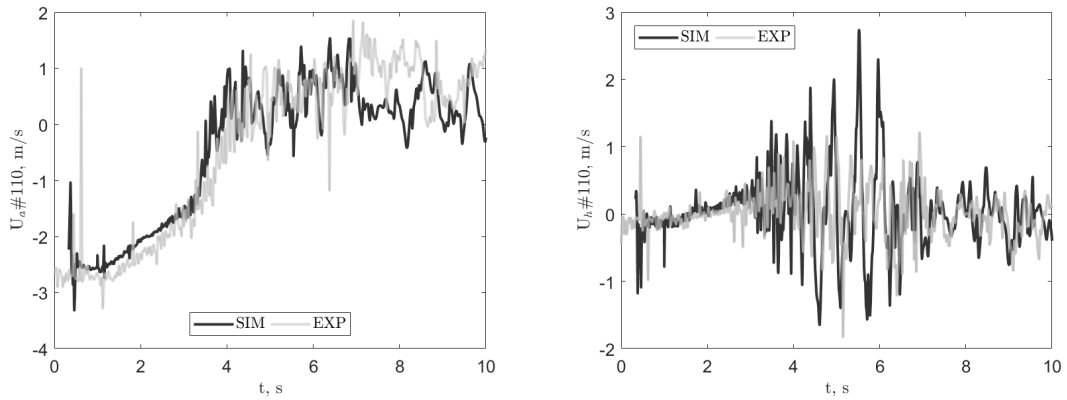
Figure 4.9: Static and fluctuating pressure in two probes during shutdown.

4.3.2 Velocity Lines

Regarding the velocity lines good and comparable results have been attained via OpenFOAM. It should be noted that the coordinate system is in negative z -direction away from the runner. This means that axial velocity moving away from the runner is negative and vice versa.

4.3.2.1 Velocity Line 1

At the velocity line closest to the runner, line 1, it can be seen that some time will pass before the guide vane changes is felt down stream. This due to that the flow does not start to decelerate at $t = 1$ s. By studying Figure 4.10a it is nice to see that the rapid deceleration of the fluid, between $t \approx 1.5 - 4.5$ s, is nicely captured by the simulation. The time interval corresponds to a guide vane angle $\alpha = 9.2^\circ - 5.3^\circ$. Worth noting is that the simulated axial velocity is unfiltered. This is not the case for the horizontal velocity in Figure 4.10b, where a filter width of 20 is used to match the experimental. The trends are well captured for both velocity components, but the fluctuations seem to be somewhat over predicted in the simulation, especially in the horizontal direction. This may be due to the reasoning from section 4.1.2, i.e. that the swirl leaving the runner is larger in the simulation than in the experiment. A large unsteady part is seen for the horizontal velocity after the rapid deceleration in axial direction.



(a) Axial velocity at position 10/28

(b) Horizontal velocity at position 10/28

Figure 4.10: Axial and horizontal velocity during shutdown for velocity line 1, at point 10 of 28.

Contour plots of axial velocity during the shutdown sequence at velocity line 1 are found in Figure 4.11. Here the simulated result is filtered with a width of 15 and only every 50th time step is showed. This to make the plot less noisy and more comparable to the experimental. The general trends are captured over the entire line, but the simulations shows more diffusive transitions between different regions. The difference under the runner seen for the steady case is also visible here, compare values in the draft tube center for $t < 2.5$ s. A tendency to generate larger turbulent structures is produced by the simulation. The draft tube mesh is quite coarse which may explain the diffusive behaviour. The filtering of the simulation is known, but no such information regarding the experiment could be found. This can perhaps explain some differences, the experimental result is almost too good to not having been filtered in some way.

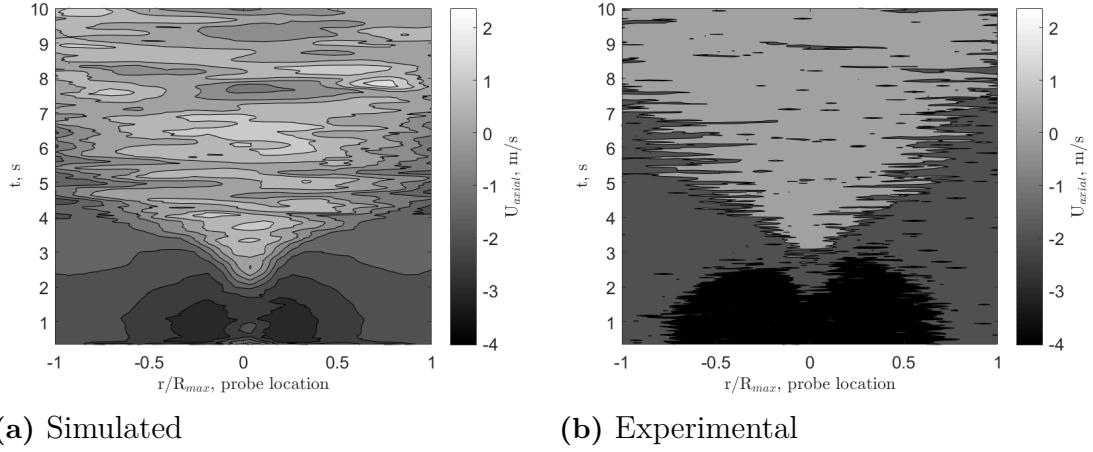


Figure 4.11: Contour plot of axial velocity during shutdown at line 1. Note that time is displayed on the y -axis and that the figure is read from bottom to top.

4.3.2.2 Velocity Line 2

As one moves further down the draft tube, to velocity line 2, it seems that the trends are even better captured, compare Figures 4.10 and 4.12. It should be noted that the data in these figures originates from probes located at different horizontal positions, but the statement is still true. The filtering properties are the same as for line 1. Thus, it can be concluded that the fluctuations are smaller at line 2 and a bit better in line with the experimental measurements than at line 1. It is also satisfying that OpenFOAM captures the messy behaviour from $t > 3$ s, for the horizontal component.

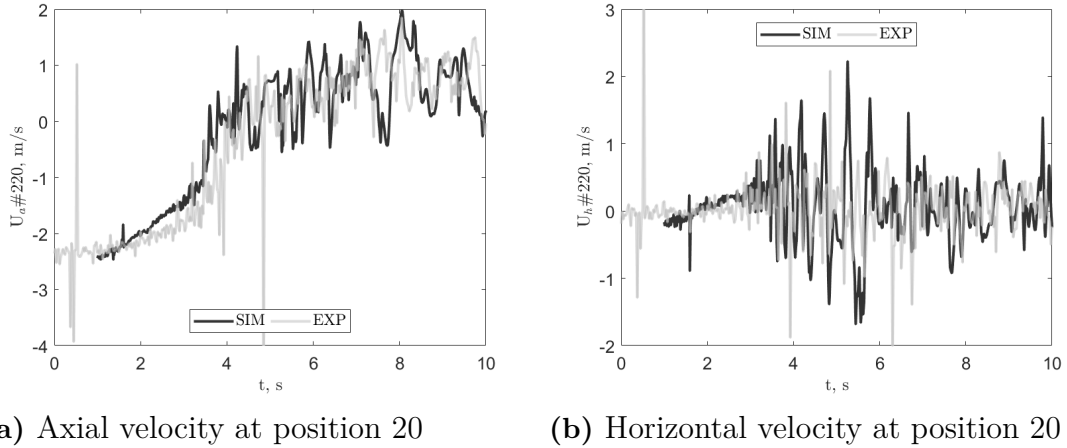


Figure 4.12: Axial and horizontal velocity during shutdown for velocity line 2, at point 20 of 28.

The velocity contour plots depicted in Figure 4.13 further emphasize that the velocity is better predicted further away from the runner. The shape of the contours at line 2 have more similarities than at line 1. The simulation still shows smeared transitions, but it coincides quite well with the experiment. It is nice to see that

part of the velocity, at the ends of the line, actually flows away from the runner at later times, $t > 5$ s. It would have been even more interesting if line 2 (and 1) spanned the entire width of the draft tube. The same filtering was used for the velocity contour plot in velocity line 2 as for line 1.

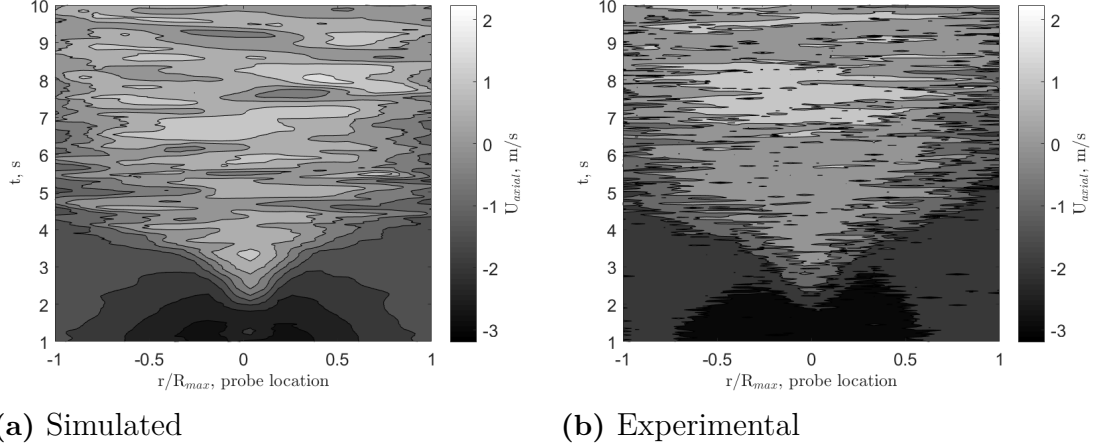


Figure 4.13: Contour plots of axial velocity during shutdown at line 2. Note that time is displayed on the y -axis and that the figure is read from bottom to top.

4.3.2.3 Velocity Line 3

Velocity line 3 is placed vertically under the runner, going towards it, i.e. point 1 is furthest away from the runner. Figure 4.14 illustrate how point three behaves during shutdown. The point is further away from the runner than line 2. The same can be stated as for line 1 and 2, the general pattern coincides well between experiment and simulation.

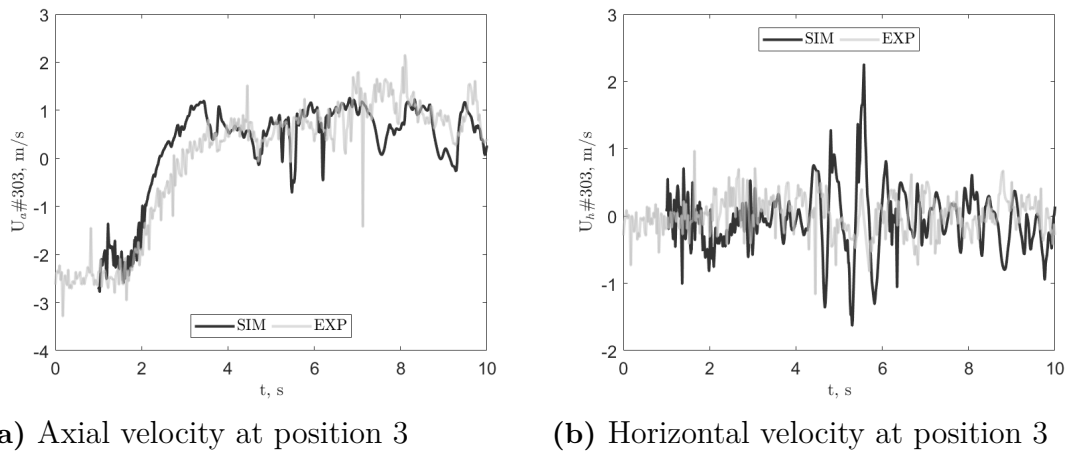


Figure 4.14: Axial and horizontal velocity during shutdown for velocity line 3, at point 3 of 19.

Contours for the axial velocity at the vertical line can be seen in Figure 4.15. The figure gives a good illustration of when the velocity goes in the direction of the draft

tube, negative velocity, and when it turns. The transition seems to occur almost at the same time regardless of vertical position. The change coincides with the start of the fast accelerations seen at lines 1 and 2, Sections 4.3.2.1 and 4.3.2.2. The simulated data is presented and filtered in the same way as in Figures 4.11 and 4.13. Experimental data was not available for all points at line 3. Hence Figure 4.15 only shows data for the first 18 probes, i.e. all except the one closest to the runner.

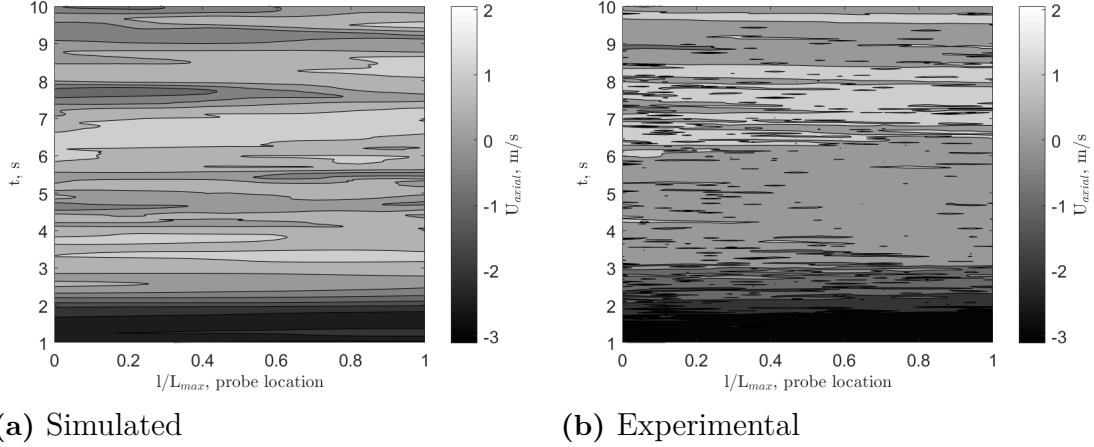


Figure 4.15: Contour plot of axial velocity during shutdown at line 3. $l/L_{max} = 0$ represents the position closest to the runner and $l/L_{max} = 1$ the furthest. Note that time is displayed on the y -axis and that the figure is read from bottom to top.

4.3.3 Shaft Thrust and Torque

Figure 4.16 visualize axial force and torque, with respect to the global origin, experienced by the runner. The quantities are computed via the OpenFOAM extension **forces**. The thrust in Figure 4.16a is filtered using a filter width of 20 and the moments in Figure 4.16b a width of three. The spikes present in both figures originates from guide vane mesh changes. The axial torque decrease linearly to almost zero Nm, while the axial thrust exhibits a small decrease but is generally constant at around 10 kN. A strange appearance at $t = 8$ s is spotted for the force, this originates from when the guide vanes have reached $\alpha = 0.80^\circ$ and the motion is altered.

Moments around x and y are comparably small, as they should be in a well designed machine. However, some interesting phenomena can be seen. If one investigate Figure 4.17, a zoomed in version of 4.16b, large unsteady behaviours in M_x and M_y are observed from $t \approx 4 - 6.5$ s. These times correlate well with the unsteadiness seen for the horizontal velocity components, right after the fast deceleration of axial velocity, in Section 4.3.2. The abrupt changes in M_x and M_y would subject the shaft bearings to unwanted fatigue loads, which might deteriorate them and thus the entire machine.

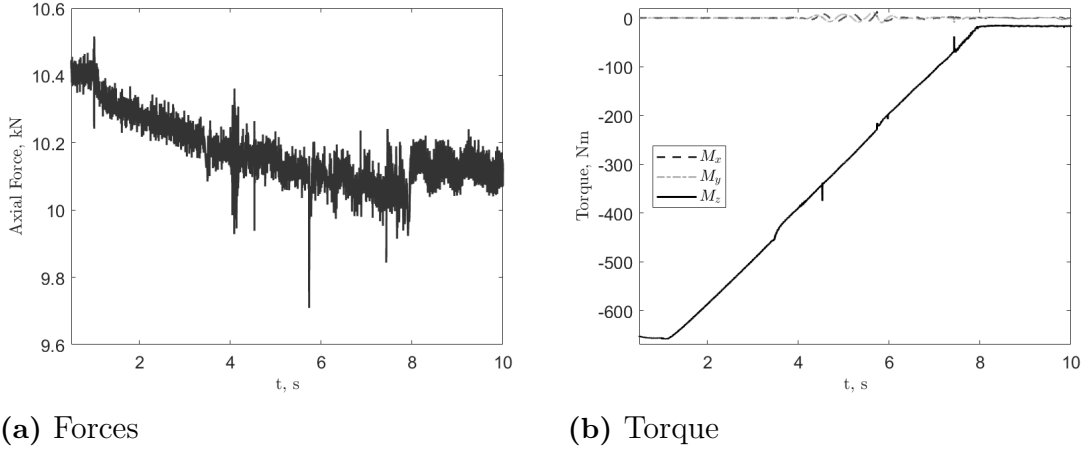


Figure 4.16: Axial thrust and torque subjected to the runner during shutdown. Note the fluctuations of M_x and M_y between $t \approx 4 - 6$ s.

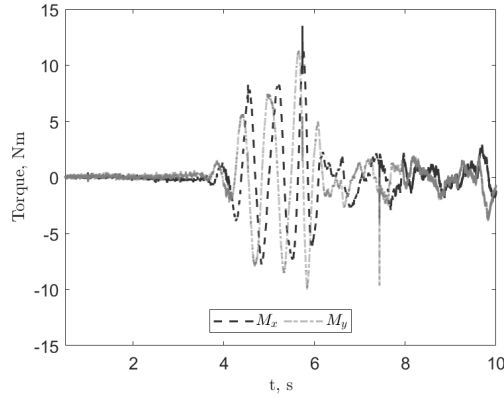


Figure 4.17: Zoomed in view of x and y moments during shutdown.

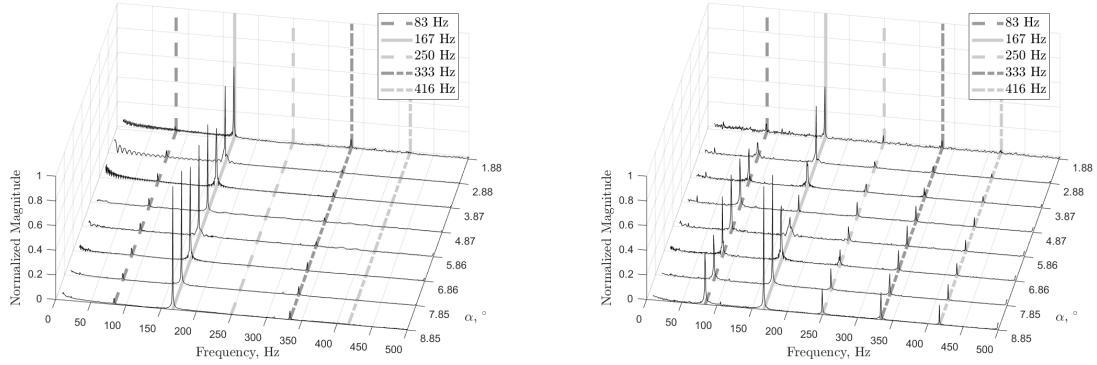
4.3.4 Frequency Spectrum

FFT can be used to reveal repeating patterns in a signal over time. When looking at the frequency spectrum of the collected data there are a number of interesting conclusions to be drawn. This because if the patterns and eigenfrequencies of a system are similar they can excite one-another and hence lead to devastating oscillations or even total failure of the machine.

Figure 4.18 depicts the frequency spectrum for pressure probes VL2, located between guide vanes and runner, and P42, connected to a runner blade. There are clear similarities and differences between the probes. Both have their peak values at 167 Hz for most guide vane angles. It is perhaps no coincidence since that value represents number of runner blades, $N_{blades} = 30$, times runner revolutions per second, $f_{runner} = 5.55$ Hz, i.e. $f_{runner}N_{blades} = 167$ Hz. Therefore, it can be stated that the speed of the runner and number of runner blades have a decisive impact of the pressure oscillations in the system. The appearance of the 167 Hz peak for P42 is

a bit peculiar as it is constant, decreasing and finally increasing to a constant value again. At VL2 it is another story, the peak has an almost sinusoidal appearance. Why it is like this or if it is due to the shutdown speed can not be determined and needs to be investigated further.

The other vital peaks seen in Figure 4.18 are all connected to the runner frequency, f_{runner} , and number of runner blades, N_{blades} . Interesting is that the probe in the vane less space, VL2, do not seem to get as effected by the harmonics as P42. Compare the peaks present at 83 Hz, as they are hardly noticeable for VL2. There are no spikes connected to the double or half frequencies, i.e. 250 Hz, 333 Hz, or 416 Hz, for probe VL2. The dominating frequency outside the runner is perhaps just 167 Hz. Hence, one must ensure that no system eigenfrequency coincides with this.



(a) Frequency spectrum for pressure probe VL2.

(b) Frequency spectrum for pressure probe P42.

Figure 4.18: Frequency spectrums of two pressure probes. $1/2 \cdot f_{runner}N_{blades} = 83$ Hz, $f_{runner}N_{blades} = 167$ Hz, $3/2 \cdot f_{runner}N_{blades} = 250$ Hz, $2 \cdot f_{runner}N_{blades} = 333$ Hz, $5/2 \cdot f_{runner}N_{blades} = 416$ Hz. Each interval is normalized by the maximum value of the entire transient.

One can also investigate the frequencies for the forces. Figure 4.19 display the frequency spectrum for the axial force acting on the runner shaft. The diagram does not display any obvious trends. This is likely an effect of the sampling frequency being too low. The force is sampled with approximately 300 Hz while the pressures and velocities are sampled ten times as often. The Nyquist theorem states that frequencies higher than half the sampling frequency are not valid [34]. This means that frequencies > 150 Hz are excluded in this case. It is plausible that if the force would have been sampled more often, it should notice some runner dynamics, as for the pressure probes.

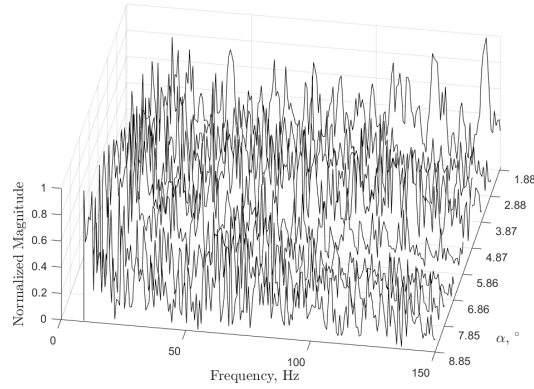


Figure 4.19: Frequency spectrum of the axial force experienced by the shaft. Each interval is normalized by its maximum value.

4.3.5 Field Plots

Paraview is used to visualize the velocity magnitude field and the Q -criterion. The snapshots have been chosen since they represent interesting phenomena during the shutdown sequence.

4.3.5.1 Velocity Field

The difference in velocity field during shutdown of the turbine can be seen in Figure 4.20. While initially showing the highest velocity in the center of the draft tube, the flow transforms and later appear to be highest at the wall. The flow in the center of the draft tube is where large differences are spotted as it goes from a velocity magnitude of approximately four m/s to zero m/s over the shutdown scenario.

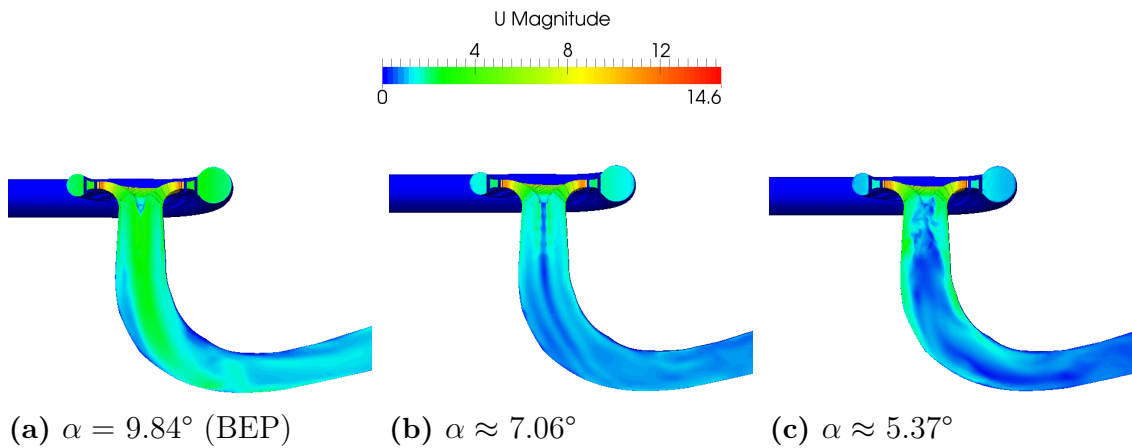


Figure 4.20: Side view of velocity magnitude field at three guide vane angles during shutdown.

The cause to why the velocity is kept at the wall might have several reasons. One is that as the mass flow decrease, thus the average velocity decrease. Meaning that a water molecule will be located in the runner for a longer period of time, thereby

being subjected to centrifugal force. The more the velocity is decreased, the more time the water have in the runner. Another aspect is perhaps that it has to do with the guide vane angle. The radial velocity component entering the runner will be reduced due to that the guide vane angle decrease and hence direct the flow in a more tangential direction. This might mean that the water at some point will lack the energy it takes to follow the hub surface of the runner in an efficient manner.

4.3.5.2 Q-Criterion

If studying the vortex rope by using an iso-surface and the Q -criterion, Figure 4.21, one can see a significant change in behaviour. The initial small rope induced in the center of the draft tube indicates a stable flow with a high axial and a moderate tangential velocity. As the velocity decrease, the flow will be concentrated to the wall as mentioned in Section 4.3.5.1. This creates a wider vortex street in the center of the draft tube and finally breaks the rope into several pieces, indicating a much more turbulent flow. The destruction of the rope happens between what is seen in Figures 4.21b and 4.21c.

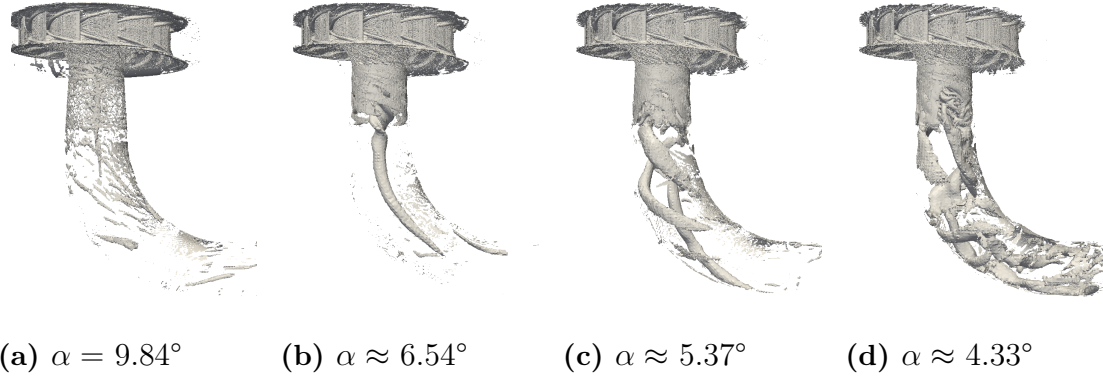


Figure 4.21: Iso-surfaces of the Q -criterion during shutdown at four guide vane angles depicting the destruction of the vortex rope, Q has an iso-value of 200. Note that the volumetric flow rates are at BEP, 67%, 55%, and 44% of BEP.

4.3.6 Comments About the Shutdown Results

LES has through this section been proven as a reliable tool to simulate the shutdown procedure. Both pressure and velocities are compared with experimental values and the general behaviors are coinciding to a satisfying extend in most cases. It is however showed that a mesh change gives a direct impact on the results via the sudden peak values present in several plots. Also, the final guide vane mesh introduced some large fluctuations which had to be filtered. This may increase the uncertainty of the results attained for $\alpha < 1.7^\circ$; $t > 7.3$ s. But the experimental measurements also lack in accuracy due to air bubbles at lower guide vane angles, making it hard to judge which is the most accurate, the numerical simulation or the experimental test case.

A pulsating behavior is encountered for $\alpha \approx 5.9 - 2.7^\circ$, $t \approx 4 - 6.5$ s, in several plots. It plausibly originates from the destruction of the vortex rope visualized in Section 4.3.5.2. Interesting to see is that M_x and M_y suddenly increases and begins to fluctuate around zero, Section 4.3.3, indicating that the shaft has to sustain radial load fluctuations on top of the axial loads. These radial effects are small in comparison to the axial torque, but they can be harmful due to fatigue and deterioration of the shaft bearings. It is uncertain whether these fluctuations would remain in a steady case at this operating point or if it is purely transient behavior.

The frequency spectrums reveals that a frequency directly linked to the runner revolution per second and number of runner blades, 167 Hz, are pulsating through the domain. This while a probe inside the runner is also subjected to multiples of the 167 Hz frequency. The eigenfrequencies of the geometry should be kept as far from these as possible. Nothing can be stated about the thrust, but it is perhaps possible to attain peaks if one utilize a higher sample rate.

The fast deceleration in the beginning of the shutdown sequence along with the destruction of the vortex rope are the most interesting phenomena during the shutdown sequence. A more conclusive study is however required to determine what they are governed by. Is it only the guide vane angle, i.e. the flow rate, that decides when these occur or is the guide vane closing speed also a contributing factor?

4.4 Start-Up Transients

Four start-up scenarios are investigated with different start-up speeds, i.e. the opening angular velocity of the guide vanes. The cases are $0.975^\circ/\text{s}$, $1.30^\circ/\text{s}$, $1.95^\circ/\text{s}$ and $2.60^\circ/\text{s}$. Figure 4.22 depicts the different opening speeds. Note that most quantities in this section are plotted against guide vane angle, α , and not time. This to make the results comparable. The final 0.80° guide vane mesh is able to open the guide vanes to about 2.70° before it crashes. Numerical data in this section is shown up to an opening angle of 2.6° to avoid errors due to bad mesh quality.

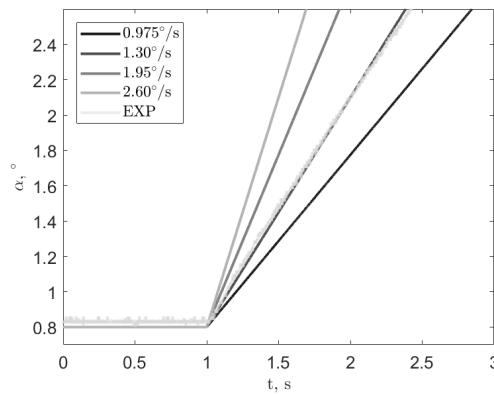


Figure 4.22: Guide vane angle change during the four start-up cases.

4.4.1 Pressure Probes

The static pressure attained at VL2 during the start-up cases follow each other and the experimental data to a great extent, see Figure 4.23a. The same may be stated for the magnitude of the pressure fluctuations at DT5, Figure 4.23b. Not much can be stated about the behavior of the static pressure at VL2, except that it seems to be increasing at the same rate regardless of opening speed. The pressure fluctuations however seem to have traces of a cyclic appearance, especially the faster opening cases along with the experimental.

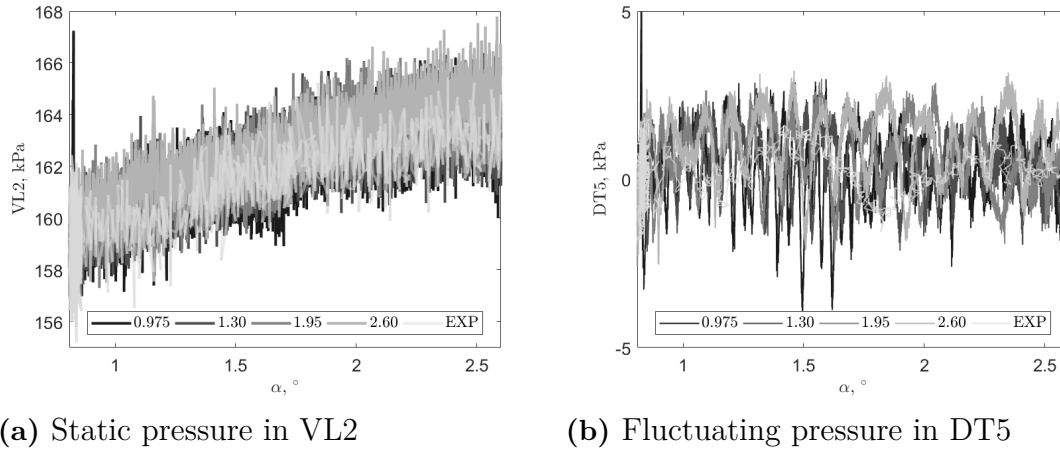


Figure 4.23: Static and fluctuating pressure at VL2 and DT5 during the start-up scenarios. The legends 0.975 - 2.60 refers to the opening speeds. Note that the numerical pressures are filtered using a width of three and five, for VL2 and DT5 respectively.

4.4.2 Velocity Line 2

Simulated data is extracted at all velocity lines for the start-up cases, but due to result similarities only results from velocity line 2 are presented. Figure 4.24 illustrates unfiltered axial and horizontal velocities at position 10 of 28 on velocity line 2. Both components display a highly unsteady behavior and for once they are in the same order of magnitude. This is perhaps not good in an efficiency point of view since there have to be a lot of swirl leaving the turbine in such a case. This does however correlate well with what was observed during the shutdown case. There will at some point be a large acceleration of the fluid in the axial direction, it would have been interesting to see if it occur at approximately the same angle regardless of opening speed or not. But due to that it happens at larger angles, it is not possible to simulate that far in the present study, the phenomena cannot be captured.

Velocity contour plots are also produced for the axial velocity at velocity line 2, see Figure 4.25. According to the contours, it seems that a slower opening speed produces a more unsteady flow than a fast. This since there are less sharp transitions between regions. Note the the data illustrated in Figure 4.25 is unfiltered. Hints of this trend could also be seen in Figure 4.24a due to that the values are fluctuating

less for the 2.60 °/s case than for 0.975 or 1.30 °/s cases. Thus, if one neglects all other parameters and only focus on velocity, a slower opening might generate more turbulence (i.e. losses) than a quick one. It could therefore be preferable to open the turbine faster than in the experiment. Whether the turbine manages this from a mechanical point of view remains to be evaluated.

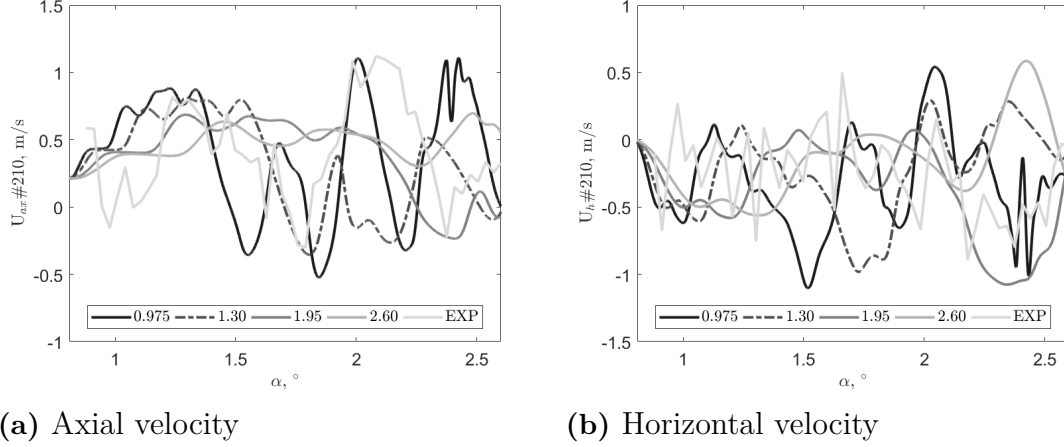


Figure 4.24: Axial and horizontal velocities at velocity line 2, point 10 of 28, during the four start-up cases.

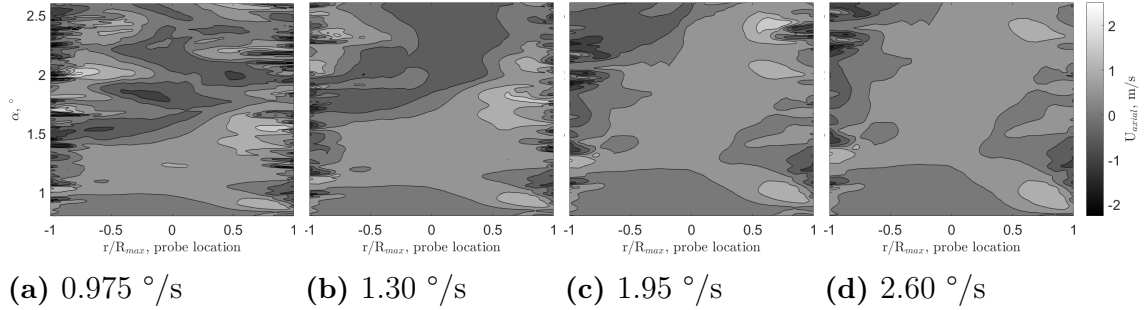


Figure 4.25: Axial velocity contour plots for the four start-up scenarios along velocity line 2. Note that time is displayed on the y -axis and that the figure is read from bottom to top.

4.4.3 Shaft Thrust and Torque

Results for axial thrust and torque induced by the fluid on the runner during the start-up cases are shown in Figure 4.26. Not much interesting can be stated about the axial torque, it simply increase linearly with the guide vane opening regardless of angular velocity. The thrust in Figure 4.26a have some distinct features. Firstly it is a bit larger with faster opening, but more interestingly, it seems to have a pulsating behavior. This appearance gets faster with slower opening speeds and may originate from that the runner rotates with a certain frequency, and since the slower opening takes longer time more cycles occur during the transient. The spike

present at $\alpha \approx 1.2$ s, for 2.60 °/s case, can not be explained and it is strange that it is there. Another even faster opening case would be good to determine whether it is a numerical error or something physical. Moments M_x and M_y are also calculated, but they are practically zero during the transient for all cases and hence not demonstrated.

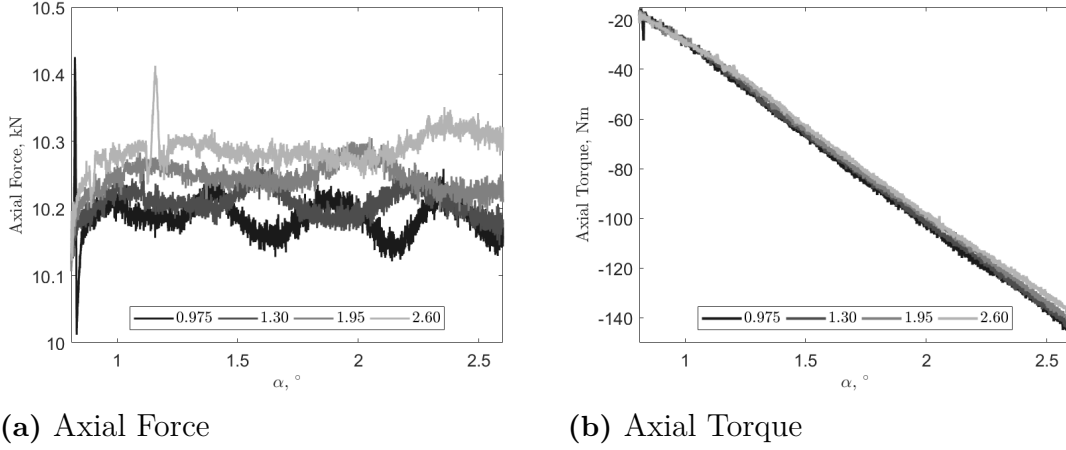


Figure 4.26: Axial force and torque subjected to the runner during the start-up cases.

4.4.4 Frequency Spectrum

By investigating the frequency spectrums attained via pressure probes VL2 and P42 one realize that there are not much difference between the start-up cases, see Figure 4.27. Essentially the same can be stated as when investigating the spectrum during shutdown, Section 4.3.4. At VL2 it is the runner revolution, $f_{runner} = 5.55$ Hz, and the number of runner blades, $N_{blades} = 30$, that are responsible for the dominating frequency, 167 Hz. P42 is effected by multiples of 167 Hz, hence the runner itself is perhaps more sensitive to oscillations.

Frequency spectrums for the axial thrust, F_z , and the force in x-direction, F_x , are seen in Figure 4.28. During the start-ups the sample frequency is increased, in conjunction with that of the shutdown transient, to about 3 kHz. This wider spectrum reveal peaks that were not visible for the shutdown case in Section 4.3.4. As can be seen for the axial force, a frequency correlated to $2 \cdot f_{runner} N_{Blades}$ is the dominating. The force F_x , acting radially, is further excited at the at the regular 167 Hz frequency as well. It would have been better for the turbines life time if no peaks where present for F_x . This since it indicates that the shaft might start to oscillate radially and thus perhaps fatigue is a risk. The 333 Hz peak seem to decrease with a faster start-up speed, whether this is true is hard to judge since the slower contain more samples.

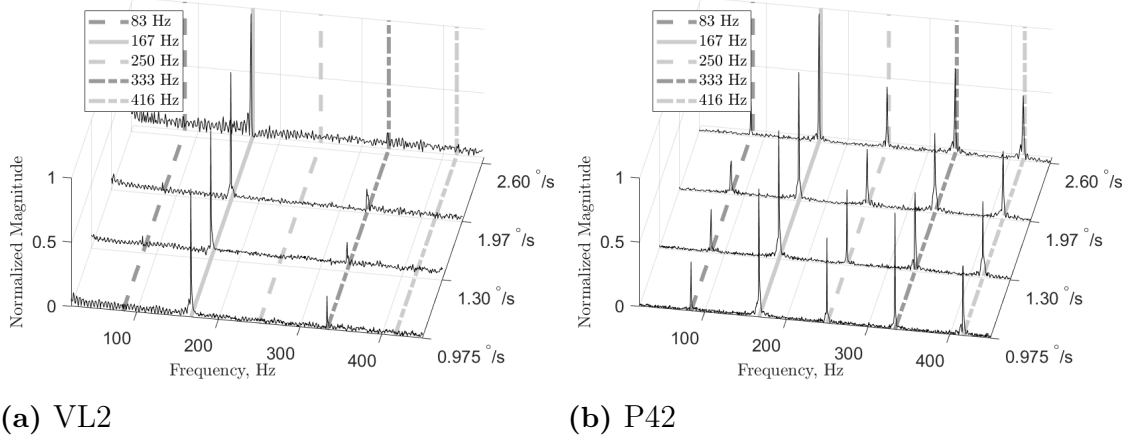


Figure 4.27: Frequency spectrum for VL2 and P42 during start-up for the four cases. $1/2 \cdot f_{runner}N_{blades} = 83$ Hz, $f_{runner}N_{blades} = 167$ Hz, $3/2 \cdot f_{runner}N_{blades} = 250$ Hz, $2 \cdot f_{runner}N_{blades} = 333$ Hz, $5/2 \cdot f_{runner}N_{blades} = 416$ Hz. Each interval is normalized with the maximum in each of the cases.

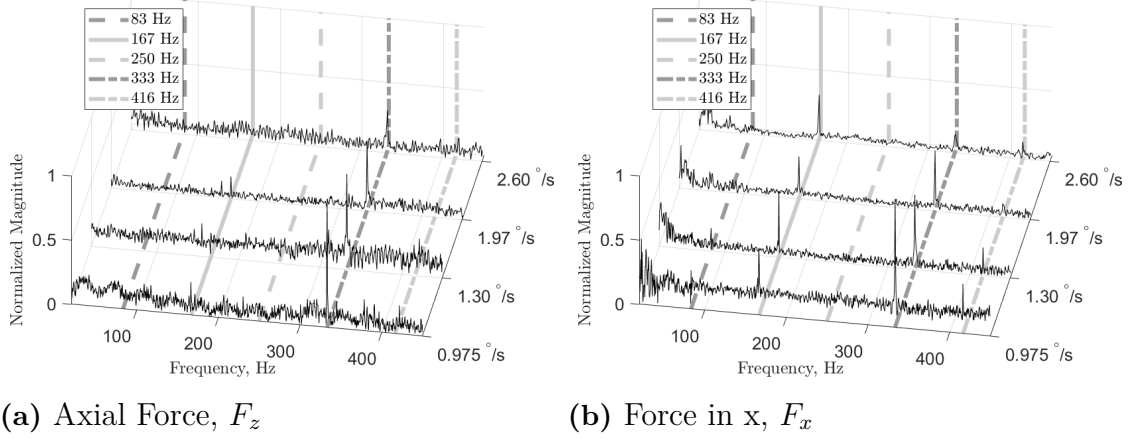


Figure 4.28: Frequency spectrum for F_z and F_x during start-up for the four cases. $1/2 \cdot f_{runner}N_{blades} = 83$ Hz, $f_{runner}N_{blades} = 167$ Hz, $3/2 \cdot f_{runner}N_{blades} = 250$ Hz, $2 \cdot f_{runner}N_{blades} = 333$ Hz, $5/2 \cdot f_{runner}N_{blades} = 416$ Hz. Each interval is normalized by the total maximum value.

4.4.5 Field Plots

Both velocity field and the Q -criterion are examined for the start-up cases. Results from them are found in this section. All snapshots are from a time step when all cases have approximately the same guide vane angle, $\alpha \approx 2.36^\circ$.

4.4.5.1 Q-Criterion

The vorticity, calculated via the Q -criterion, seems to be rather indifferent to guide vane opening speed at first glance. But if one studies Figure 4.29 closely one realize

that there are differences just beneath the runner. The structures appears less violent as the guide vane opening speed increases. The result goes well inline with what is already seen in the velocity contour plots in Figure 4.25. This further emphasizes the hypnotizes stated in Section 4.4.2, i.e. that from a flow point of view perhaps a faster start-up is preferable to a slow since the slower generates larger flow structures and thus losses. Further downstream, the structures have similar appearance regardless of opening speed.

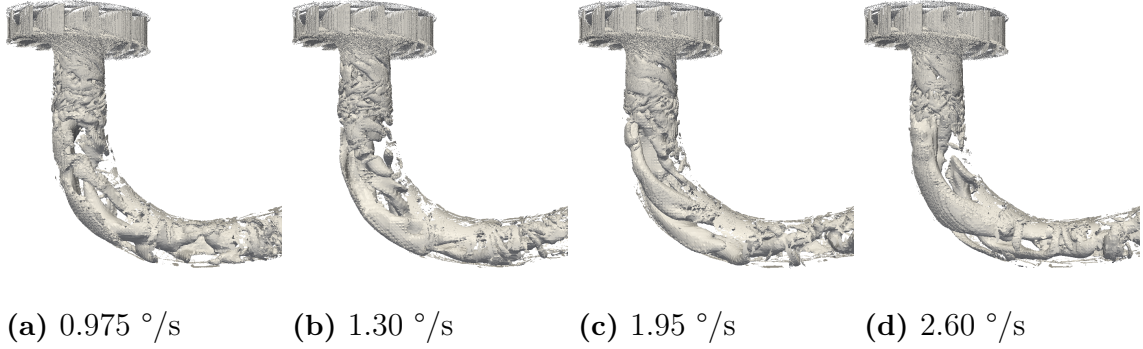


Figure 4.29: Iso-surface of the Q-criterion at $\alpha \approx 2.36^\circ$ for all start-up cases. The iso-value is set to 200.

4.4.5.2 Velocity Field

An explanation to why a faster start-up scenario seemed preferable earlier is perhaps found in Figure 4.30, note that it is just snapshots in time. The interesting phenomena happen just under the runner, as the flow is quite similar downstream of the draft tube between cases. The fluid velocity is more centered to the tube wall in faster cases than in slower. This may be due to that the flow have had longer time to adjust to the increased flow rate.

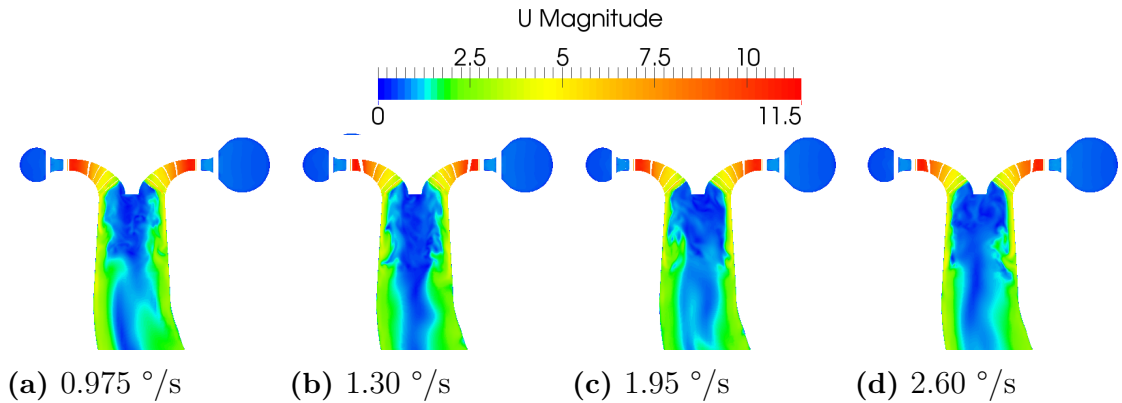


Figure 4.30: Side view of velocity magnitude at $\alpha \approx 2.36^\circ$ for all start-up cases.

Another scenario is the other way around, a slow start-up is preferable. Perhaps the more turbulent buildup is due to that the vortex rope creation occur at a lower

guide vane angle, and what is seen in Figure 4.30 is just that. As found during the shutdown case in Section 4.3, the destruction of the vortex rope increase the radial moments subjected to the shaft. It is probably preferable to have a vortex rope creation at a lower α if possible since less water is in motion at this point.

4.4.6 Comments About the Start-Up Results

Firstly, it can be concluded that the pressure in the turbine show low sensitivity to how fast the guide vanes are opened. The differences are instead in how the velocity field is developing. The highly swirling flow in the beginning of the draft tube can be explained by the low flow rate in these early stages of the opening. The interesting part is the way the slower opening speeds display a more turbulent flow, and since this is associated with larger losses it might be desirable to avoid these. As long as the machine can manage a faster start-up this seems more promising due to that it takes shorter time, exhibits fewer flow structures, and according to Section 4.4.3 the shaft is subjected to practically the same axial loads regardless of opening speed.

The alternative hypothesis, contradictory to the previous, is that the slower case has already started the creation of the vortex rope. The high turbulence could be associated with a changing state as indicated in Section 4.3.5.2. This would imply that the rope is not just only a function of guide vane angle but also of opening speed. As seen for the shutdown case, Figure 4.17, the moments in x and y becomes unstable during the destruction of the vortex rope. The same is presumably true for the creation of it. The unsteadiness in x and y would plausibly deteriorate the shaft bearings and thus the whole turbine. It is quite safe to assume that the rope creation is desirable at a lower guide vane angle since less water and hence forces act on the turbine at that state. Thus, whether fast or slow opening speeds are better can not be concluded. A more conclusive investigation is required to determine the root cause of the vortex rope creation.

All in all, it would have been good to be able to run the start-up simulations longer to capture the creation of the vortex rope. This to see if a more decisive conclusion can be drawn whether a faster case really is better than a slow, or vice versa. By just looking at the torque and axial force and combining them with other sections from the start-up suite one can not state what kind of start-up that is preferable.

5

Conclusions

The aim of the study was to simulate and verify shutdown and start-up transients of the Francis-99 hydro turbine using CFD in OpenFOAM. The simulations are performed using LES and several mesh setups. One steady BEP case, an entire shutdown sequence and four short start-up scenarios have been produced and evaluated. The verification part contains comparisons between numerically computed velocities and pressures with values from experimental test cases conducted by NTNU.

It is with confidence concluded that the simulation procedure generates trustworthy results. This since the produced results are not only comparable with the experimental, they do show the same behaviour and close to the same value in several cases. Hence LES along with the dynamic eddy viscosity model are deemed as highly suitable to simulate and spot trends for at least steady conditions and shutdown scenarios. No complete start-up sequence is evaluated. But it is possibly safe to assume that it also captures the flow appearance to a great extent since the short start-up suite indicates this.

In the beginning of this study it was hoped that the results would provide an answer to how an optimal start-up and/or shutdown sequence should be performed. Nothing about this can be stated since only one shutdown case has been evaluated and the start-up scenarios were too short to really capture the decisive phenomena, i.e. the creation of a vortex rope and the fast acceleration following. It can however be stated that the axial torque, subjected to the runner, varies linearly with the guide vane opening regardless of motion speed. It is further concluded that the opening speed has an impact on the development of the flow in the draft tube. But to confidently state anything about optimal procedure more and longer simulations are required.

A drawback with the procedure is that it demands vast computational resources. In general, the simulations were produced using a large cluster containing 400 cores. The entire shutdown sequence took about one and a half month to simulate. That duration includes the development of the steady flow before the transient commenced. Another comment about the results is that at lower guide vane angles, the pressure probes in both the draft tube and the vane less region slightly deviate from the experimental. Whether this has to do with low mesh quality, errors in the simulation, or uncertainty in the experiments due to air bubbles at small angles is hard to conclude. As a first step it is advisable to run the simulations on a finer mesh to ensure mesh independent results.

6

Further Work

The performed study stands as an optimal starting point to do further and deeper analysis. A desire on the behalf of Vattenfall was that this thesis work should provide answers how to optimally start-up and shutdown a Francis hydro-turbine based on loads on the runner. A natural recommendation is thus to use this platform to conduct more transient simulations to do just that. While doing so the creation and destruction of the vortex rope is of interest. Some effort should be put into analysis and preferably describing what governs the rope. Is it only the guide vane angle, i.e. volumetric flow rate, or is it also a function of guide vane angular velocity.

The performed work do not contain any direct mesh dependency study. It is known that the mesh is quite coarse, especially in the draft tube. Hence, it would be preferable to ensure that the results are independent of the mesh. A solution to this is to perform a mesh dependency study, perhaps not for the entire transients, but at least at some steady condition. It was found that the final guide vane mesh produced a more noisy solution, if this only was due to mesh quality could not be concluded, hence one should try to use a finer mesh for the lower guide vane angles.

A more comprehensive Fluid Structural Interaction (FSI) case could be highly interesting to perform. As a start one should try to use a six degree of freedom solver for the runner motion. This to validate if the energy of the water can make the runner rotate, without speed fluctuations, at the required RPM. Also of interest is how to simulate the acceleration of the runner from stand still. A further step would be to perform full FSI where runner blades and shaft deformation are taken into consideration due the forces they are subjected to.

A final recommendation would be to make the model less complex. The model is in present state very computationally demanding and require vast resources. To really perform an optimization study, faster models are preferable since several different cases needs to be investigated. It is naturally impossible to decrease the complexity and still uphold the same level of accuracy. It is perhaps sufficient if a simpler model can capture trends rather than the exact numbers to find more optimal start-up and shutdown procedures.

Bibliography

- [1] Dewan, Y., et al. '*Simulation of the Francis-99 Hydro Turbine During Steady and Transient Operation*', , Journal of Physics: Conference Series, vol. 782/no. 1, 2017.
- [2] Francis-99 Committee, '*Norwegian Hydropower Center*', [Website] <https://www.ntnu.edu/nvks/f99-test-case2>, (accessed 2018-05-29).
- [3] Karlsson, T., '*Luleå Tekniska Universitetet*', [website], 2014, <https://www.ltu.se/research/subjects/Stromningslara/Konferenser/Francis-99/Test-Case-1.111520>, (accessed 2018-02-14).
- [4] Mössinger, P., et al., '*Francis-99: Transient CFD Simulation of Load Changes and Turbine Shutdown in a Model Sized High-Head Francis Turbine*', Journal of Physics: Conference Series, vol. 782/no. 1, 2017.
- [5] Diagne, I., et al., '*Modeling the dynamic behavior of turbine runner blades during transients using indirect measurements*', IOP Conference Series: Earth and Environmental Science, 2016
- [6] Dixon S. L., Hall C., '*Fluid Mechanics and Thermodynamics of Turbomachinery*', 7th edition, UK, Butterworth-Heinemann, 2013
- [7] Stonecypher, L., '*Bright Hub Engineering*', [Website], 2009, <https://www.brighthubengineering.com/fluid-mechanics-hydraulics/27407-hydraulic-turbines-francis-turbine/>, (accessed 2018-04-04).
- [8] Trivedi, C., Cervantes M.J., et al., '*Experimental and Numerical Studies of a High-Head Francis Turbine: A Review of the Francis-99 Test Case*', Energies, vol. 9/no. 2, 2016, pp. 1-24.
- [9] Trivedi, C., Cervantes, M.J., Dahlhaug, O.G., '*Francis-99: A test-case on a high head Francis turbine*', Waterpower laboratory, Department of Energy and Process Engineering, Faculty of Engineering, NTNU – Norwegian University of Science and Technology, Trondheim, Norway, 2019.
- [10] Davidson L., '*Fluid mechanics, turbulent flow and turbulence modeling.*', Unpublished, 2017.
- [11] Versteeg H., Malalasekera W., '*An Introduction to Computational Fluid Dynamics The Finite Volume Method*', USA, Pearson Education, 2007
- [12] Moukalled, F., L. Mangani, et al., '*The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*', Anonymous Translator, Volume 113, 1st 2016.;1st 2015; edn, Cham, Springer International Publishing, 2016.
- [13] Ferziger, J.H., and Peric, M., '*Computational Methods for Fluid Dynamics*', 3., rev. edn, Berlin, Springer, 2002.

- [14] Rieth, M., et al. '*Comparison of the Sigma and Smagorinsky LES Models for Grid Generated Turbulence and a Channel Flow*', Computers & Fluids, vol. 99, 2014, pp. 172-181.
- [15] Khani, S., Waite, M., '*Large Eddy Simulations of Stratified Turbulence: The Dynamic Smagorinsky Model*', Journal of Fluid Mechanics, vol. 773, 2015, pp. 327-344.
- [16] Germano, M., et al. '*A dynamic subgrid-scale eddy viscosity model*', Physics of Fluids A: Fluid Dynamics, vol. 3, no. 7, 1991, pp. 1760-1765.
- [17] Pope, S.B., '*Turbulent flows*', Cambridge University Press, Cambridge, 2000
- [18] Lillberg, E., Nilsson, P., '*Generic CFD for Vortex Induced Acoustic Resonance in Deep Cavities*', ICONE16, 2008
- [19] Sagaut, P., '*Large Eddy Simulation for Incompressible Flows: An Introduction*', SpringerLink, 2002
- [20] Bos, F.M., '*Numerical simulations of flapping foil and wing aerodynamics, Mesh deformation using radial basis functions*', Doctoral Thesis, Delft University of Technology, 2010.
- [21] Jasak, H., Tukovic, Z., '*Automatic mesh motion for the unstructured Finite Volume Method*', Transactions of FAMENA, 30. 1-20.
- [22] "The Architects of OpenFOAM", '*About OpenFOAM*', [Website], <https://cfd.direct/openfoam/about/>, (accessed 2018-02-15).
- [23] Greenshields, C., '*The OpenFOAM Foundation*', [Website], 2014, <https://openfoam.org/release/2-3-0/non-conforming-ami/>, (accessed 2018-02-14).
- [24] Behrens, T., '*OpenFOAM's basic solvers for linear systems of equations*', edited by Nilsson H., 2008.
- [25] Christopher J., '*The OpenFOAM Foundation User Guide version 5.0*', OpenFOAM Foundation Ltd, 2017.
- [26] Stoessel, L., '*Numerical Simulations of the flow in the Francis-99 Turbine*', Master Thesis, Chalmers University of Technology, 2014.
- [27] VÉSTEINSSON, G., Master Thesis, Unpublished (Internal document), Chalmers University of Technology, 2015
- [28] Holmén V., '*Methods for Vortex Identification*', Masters Thesis, Lund University, 2012.
- [29] Holzmann, T., '*Mathematics, Numerics, Derivations and OpenFOAM(R)*', Holzmann CFD, Leoben, fourth edition, 2017
- [30] Auvinen, M., et al, '*Sig Turbomachinery Library turboPerformance*', [Website], https://openfoamwiki.net/index.php/Sig_Turbomachinery_Library_turbPerformance, (accessed 2018-04-25).
- [31] Unknown, '*MathWorks*', [Website], <https://se.mathworks.com/help/curvefit/smooth.html>, (accessed 2018-05-08).
- [32] Unknown, '*MathWorks*', [Website], <https://se.mathworks.com/help/matlab/ref/fft.html>, (accessed 2018-05-14).
- [33] Unknown, '*MathWorks*', [Website], <https://se.mathworks.com/help/signal/ref/resample.html>, (accessed 2018-05-23).

- [34] Lévesque, L., '*Nyquist Sampling Theorem: Understanding the Illusion of a Spinning Wheel Captured with a Video Camera*', *Physics Education*, vol. 49/no. 6, pp. 697-705, 2014.

A

Appendix

A.1 gv_mesh_script

```
#!/bin/bash
GV=gv_2.0deg_2018_04_19
cd ../../../../scratch/dgh38/Thesis_2018/Mesh/GV/
cd gv_rotation_and_stl/
rm -r GV1_*
rm -r $GV
surfaceTransformPoints -translate '(-39.529966 -350.75955 0 )' gv_01_BEP.stl GV1_trans.stl
surfaceTransformPoints -rollPitchYaw '(0 0 7.84)' GV1_trans.stl GV1_trans_rot.stl
surfaceTransformPoints -translate '(39.529966 350.75955 0 )' GV1_trans_rot.stl GV1_rot.stl
surfaceTransformPoints -rollPitchYaw '(0 0 -12.857142857142858)' GV1_rot.stl GV_temp1.stl
sed -i s/"patch28"/"gv_01"/g GV1_rot.stl
sed -i s/"patch29"/"gv_01"/g GV1_rot.stl
surfaceAdd GV1_rot.stl GV_temp1.stl GV.stl
sed -i s/"patch28"/"gv_02"/g GV.stl
sed -i s/"patch29"/"gv_02"/g GV.stl

x=1
while [ $x -le 26 ]
do
surfaceTransformPoints -rollPitchYaw '(0 0 -12.857142857142858)' GV_temp1.stl GV_temp1.stl
surfaceAdd GV.stl GV_temp1.stl GV.stl
NoGV=$(( 2+$x ))
if [ $NoGV -le 9 ]
then
sed -i s/"patch28"/"gv_0$NoGV"/g GV.stl
sed -i s/"patch29"/"gv_0$NoGV"/g GV.stl
fi
sed -i s/"patch28"/"gv_$NoGV"/g GV.stl
sed -i s/"patch29"/"gv_$NoGV"/g GV.stl
x=$(( $x + 1 ))
done

mv GV.stl $GV.stl
echo Rotation is compleate
cd ..
cp -r GV_base_mesh/ $GV/
cd $GV/
rm -r GV_All.stl
cp ../gv_rotation_and_stl/$GV.stl .
sed -i s/"GV.stl"/"$GV.stl"/g system/meshDict
surfaceAdd $GV.stl gv_inlet.stl GV_All.stl
surfaceAdd GV_All.stl gv_outlet.stl GV_All.stl
```

```

surfaceAdd GV_All.stl gv_upper_v2.stl GV_All.stl
surfaceAdd GV_All.stl gv_lower_v2.stl GV_All.stl
cartesianMesh
transformPoints -scale '(0.001 0.001 0.001)'

```

```

echo gv mesh creation compleat

```

A.2 mergeF99

```

#!/bin/bash
temp_Mesh=0.80deg_hope_2018_04_30
GV_mesh=gv_0.8deg_Eric_hope_2018_04_30
cp -r Runner_test/. $temp_Mesh/
cd $temp_Mesh/
transformPoints -rollPitchYaw '(0 0 -252.9365)'
cd ..
mergeMeshes -overwrite $temp_Mesh/ DT_orig/
mergeMeshes -overwrite $temp_Mesh/ SV_test/
mergeMeshes -overwrite $temp_Mesh/ GV/$GV_mesh/
cd $temp_Mesh/
rm -r Allname*
rm -r VTK

```

```

echo Merging Meshes Compleat

```

A.3 case_setup

```

#!/bin/bash
case_folder=0.80deg_orig_2018_04_27
cd ../../../../scratch/dgh38/Thesis_2018/Runs/
rm -r $case_folder/system/
rm -r $case_folder/0*
cp -r case_properties/system/ $case_folder/.
cp -r case_properties/0_orig/ $case_folder/0/
cp -r case_properties/constant/. $case_folder/constant/.
cd $case_folder/
cp -r 0/ 0_orig/
changeDictionary
foamJob -s checkMesh -allTopology -allGeometry
setSet -batch ../case_properties/make_frozen_and_rotating_v2
2123myFlattenMesh
cd ../
echo Case Setup Compleat
echo Remember to uppdate rotated probes

```

A.3.1 make_frozen_and_rotating_v2

```

faceSet gv_outlet new patchToFace gv_outlet
pointSet gv_outlet new faceToPoint gv_outlet all
pointSet frozen new cellToPoint region0 all
pointSet frozen add pointToPoint gv_outlet
pointZoneSet frozenZone new setToPointZone frozen

```

```

pointSet rotating new cellToPoint region0 all
pointZoneSet rotatingZone new setToPointZone rotating
cellSet region0 new cellToCell region0
cellSet region1 new cellToCell region1
cellSet region2 new cellToCell region2
cellSet region3 new cellToCell region3
cellZoneSet region0 new setToCellZone region0 all
cellZoneSet region1 new setToCellZone region1 all
cellZoneSet region2 new setToCellZone region2 all
cellZoneSet region3 new setToCellZone region3 all

// region0 = Runner
// region1 = DT
// region2 = SC+SV
// region3 = GV

```

A.3.2 123myFlattenMesh

```

/*-----*\
=====
\\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox
\\      /  O peration  |
\\      /  A nd        | Copyright (C) 2011-2013 OpenFOAM Foundation
  \\    /  M anipulation |
-----*/

License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Application
    flattenMesh

Description
    Flattens the front and back planes of a 2D cartesian mesh.

/*-----*/

#include "argList.H"
#include "Time.H"
#include "polyMesh.H"
#include "pointBoundaryMesh.H"
#include "polyBoundaryMesh.H"
#include "fvMeshSubset.H"

```

```

#include "pointFields.H"

using namespace Foam;

// * * * * *

int main(int argc, char *argv[])
{
    # include "setRootCase.H"
    # include "createTime.H"
    # include "createPolyMesh.H"

    pointIOField points
    (
        IOobject
        (
            "points",
            runTime.findInstance(polyMesh::meshSubDir, "points"),
            polyMesh::meshSubDir,
            runTime,
            IOobject::MUST_READ,
            IOobject::NO_WRITE,
            false
        )
    );

    scalar value = 0.0297992;
    const label patchID = mesh.boundaryMesh().findPatchID("gv_upper");
    labelList ppoints = mesh.boundaryMesh()[patchID].meshPoints();
    forAll(ppoints, pointI)
    {
        points[ppoints[pointI]].z() = value;
    }

    scalar value2 = -0.0297992;
    const label patchID2 = mesh.boundaryMesh().findPatchID("gv_lower");
    labelList ppoints2 = mesh.boundaryMesh()[patchID2].meshPoints();
    forAll(ppoints2, pointI)
    {
        points[ppoints2[pointI]].z() = value2;
    }

    // Set the precision of the points data to 10
    IOstream::defaultPrecision(max(10u, IOstream::defaultPrecision()));

    Info<< "Writing points into directory " << points.path() << nl << endl;
    points.write();

    return 0;
}

// *****

```

A.4 doublePimpleDyFoam_333rpm

```
/*-----*\
=====
\\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox
\\      /  O peration  |
\\      /  A nd        | Copyright (C) 2011-2013 OpenFOAM Foundation
  \\    /  M anipulation |
-----\

License
  This file is part of OpenFOAM.

  OpenFOAM is free software: you can redistribute it and/or modify it
  under the terms of the GNU General Public License as published by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version.

  OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
  ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
  FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
  for more details.

  You should have received a copy of the GNU General Public License
  along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Application
  pimpleDyMFoam.C

Description
  Transient solver for incompressible, flow of Newtonian fluids
  on a moving mesh using the PIMPLE (merged PISO-SIMPLE) algorithm.

  Turbulence modelling is generic, i.e. laminar, RAS or LES may be selected.

\*-----*/

#include "fvCFD.H"
#include "dynamicFvMesh.H"
#include "singlePhaseTransportModel.H"
#include "turbulenceModel.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"

// * * * * *

int main(int argc, char *argv[])
{
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createDynamicFvMesh.H"
    #include "initContinuityErrs.H"

    pimpleControl pimple(mesh);

    #include "createFields.H"
```

```

#include "createUf.H"
#include "createFvOptions.H"
#include "readTimeControls.H"
#include "createPcorrTypes.H"
#include "CourantNo.H"
#include "setInitialDeltaT.H"

// * * * * *

Info<< "\nStarting time loop\n" << endl;

while (runTime.run())
{
    #include "readControls.H"
    #include "CourantNo.H"

    #include "setDeltaT.H"

    runTime++;

    Info<< "Time = " << runTime.timeName() << nl << endl;

    mesh.update();

    vector origin_(0,0,0);
    scalar omega_ = -34.8717;
    vector axis_(0,0,1);
    scalar t = runTime.time().value();

    // Rotation around axis
    scalar angle = omega_*t;

    quaternion R(axis_, angle);
    septernion TR(septernion(origin_)*R*septernion(-origin_));

    Info<< "solidBodyMotionFunctions::rotatingMotion::transformation(): "
        << "Time = " << t << " transformation: " << TR << endl;

    label pzi = mesh.pointZones().findZoneID("rotatingZone");
    const pointZone& pz = mesh.pointZones()[pzi];

    pointField transformedPts(mesh.points());

    UIndirectList<point>(transformedPts, pz) =
        transform
        (
            TR,
            pointField(transformedPts, pz)
        );

    mesh.movePoints(transformedPts);

    // Calculate absolute flux from the mapped surface velocity
    phi = mesh.Sf() & Uf;

    if (mesh.changing() && correctPhi)

```

```

{
    #include "correctPhi.H"
}

// Make the flux relative to the mesh motion
fvc::makeRelative(phi, U);

if (mesh.changing() && checkMeshCourantNo)
{
    #include "meshCourantNo.H"
}

// --- Pressure-velocity PIMPLE corrector loop
while (pimple.loop())
{
    #include "UEqn.H"

    // --- Pressure corrector loop
    while (pimple.correct())
    {
        #include "pEqn.H"
    }

    if (pimple.turbCorr())
    {
        turbulence->correct();
    }
}

runTime.write();

Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
      << "   ClockTime = " << runTime.elapsedClockTime() << " s"
      << nl << endl;
}

Info<< "End\n" << endl;

return 0;
}

// *****

```

A.5 Guide Vane Modified BC

The guide vane linear rotating bc is a modification of the existing angularOscillatingDisplacementPointPatchVectorField boundary condition. The modifications starts at line 152 in the original BC. Both the modified and the same part of the original bc are showed below, from line 152.

Modified:
{


```

    if (this->updated())
    {
        return;
    }

    const polyMesh& mesh = this->dimensionedInternalField().mesh();
    const Time& t = mesh.time();
    //    const pointPatch& p = this->patch();

    scalar angle = angle0_ + omega_*((Foam::constant::mathematical::pi)/180)*t.value();
    vector axisHat = axis_/mag(axis_);

    Info << "angle = " << angle << " deg" << endl;

    vectorField p0Rel(p0_ - origin_);

    vectorField::operator=
    (
        (
            p0Rel*(cos(angle) - 1)
            + (axisHat ^ p0Rel*sin(angle))
            + (axisHat & p0Rel)*(1 - cos(angle))*axisHat
        )
    );

    fixedValuePointPatchField<vector>::updateCoeffs();
}

```

Original:

```

{
    if (this->updated())
    {
        return;
    }

    const polyMesh& mesh = this->dimensionedInternalField().mesh();
    const Time& t = mesh.time();

    scalar angle = angle0_ + amplitude_*sin(omega_*t.value());
    vector axisHat = axis_/mag(axis_);
    vectorField p0Rel(p0_ - origin_);

    vectorField::operator=
    (
        p0Rel*(cos(angle) - 1)
        + (axisHat ^ p0Rel*sin(angle))
        + (axisHat & p0Rel)*(1 - cos(angle))*axisHat
    );

    fixedValuePointPatchField<vector>::updateCoeffs();
}

```