Graz University of Technology
Faculty of Cicil Engineering
Institute of Hydraulic Engineering
and Water Resources Management

2D and 3D Numerical Simulations of a fixed Weir using open-source Codes

Master's Thesis

submitted by

Martin Peßl, BSc

First Reviewer: Univ.-Prof. Dipl.-Ing. Dr.techn. Gerald Zenz

Second Reviewer: Shervin Shahriari, M.Sc.

Graz, November 5, 2017

Abstract

Within this master's thesis numerical simulations on a weir located in Vorarlberg, Austria, were performed. The analyses were based on a hydraulic model test performed in 2014 at the Institute of Hydraulic Engineering and Water Resources Management of Graz University of Technology. This hydraulic model test should prove the required capacity of an existing weir structure after renovation. This thesis compares the measurements done in this model test with results of Computational Fluid Dynamics (CFD). Furthermore, the performed simulations allowed additional results, not provided by the model test.

Computations were conducted with the open-source tool *OpenFOAM* using multiphase analyses. Other non-commercial tools were used for pre- and post-processing. Regarding the tools used for preprocessing, a comparison the meshing tool *Salome* and *OpenFOAM's* own meshing algorithm snappyHexMesh was performed to examine their suitability for such analyses. The used geometries were based on the already simplified physical model test.
The main focus lies on a comparison of the overflow coefficient using two and three dimensional numerical models and two different discharges. Another important point of interest was the pressure occurring along the weir. These results were evaluated using different modifications in geometries. Additionally, a sensitivity study regarding several parameters of interest was carried out. Furthermore, this thesis investigates the required aeration of this weir after renovation.

The analyses showed, that water levels in the reservoir acquired with CFD were lower than in the physical model test. The weir's capacity after renovation according to numerical simulations was quite close to the capacity of the existing weir. This desired state was not reached in the model test. In conclusion, the numerical simulations showed the same behaviour to geometrical changes made in the model test. Despite different results regarding the water level, the simulations are an important addition to a physical model test. This thesis also showed the high capabilities of open-source software.

Kurzbeschreibung

Im Rahmen dieser Masterarbeit wurden numerische Simulationen eines Wehres in Vorarlberg durchgeführt. Die Simulationen basieren auf einem hydraulischen Modellversuch durchgeführt im Jahr 2014 am Institut für Wasserbau und Wasserwirtschaft der Technischen Universität Graz. Dieser hydraulische Modellversuch sollte die Förderfähigkeit eines bestehenden Wehres nach einem Umbau bestätigen. Das Ziel dieser Masterarbeit war der Vergleich der im Modellversuch erhaltenen Messergebnisse mit Resultaten erhalten aus numerischen Modellen. Zusätzlich erlaubten die durchgeführten Simulationen auch weitere Resultate, die nicht im Zuge des Modellversuchs geliefert wurden.

Sämtliche Berechnungen wurden mit dem open-source Programm *OpenFOAM* mittels mehrphasen-Modellen durchgeführt. Für die Modellaufbereitung und die Auswertung der erhaltenen Daten wurden andere open-source Programme verwendet. Bezüglich der Netzgenerierung wurde ein Vergleich zwischen dem Programm *Salome* und *OpenFOAMs* eigenen Algorithmus snappyHexMesh erstellt, um ihre Tauglichkeit für solche Analysen zu untersuchen. Die verwendeten Geometrien basierten auf dem bereits vereinfachten physikalischen Modellversuch.
Der Hauptfokus lag auf dem Vergleich der Überfall-Koeffizienten erhalten aus verschiedenen zwei- und dreidimensionalen Berechnungen. Ein weiterer wichtiger Punkt war die Untersuchung der am Wehrkörper auftretenden Drücke. Diese Untersuchungen wurden für verschiedene Modifizierungen der Geometrie ausgewertet. Zusätzlich wurde eine Sensitivitätsanalyse des numerischen Modells betreffend einiger bedeutender Parameter durchgeführt. Zuletzt wurde noch die benötigte Belüftung nach dem Wehrüberfall untersucht.

Die Resultate bezüglich des Wasserstandes im Speicher dieser Analysen zeigten, dass die numerischen Ergebnisse eindeutig unter denen des Modellversuchs liegen. Die Förderfähigkeit des Wehres zufolge der numerischen Simulationen lag nahe an der Förderfähigkeit des bestehenden Wehres. Dieser Sollzustand wurde im hydraulischen Modellversuch nicht erreicht. Zusammenfassend zeigten geometrische Änderungen der numerischen Analysen dasselbe Verhalten wie Änderungen im Modellversuch. Trotz der Differenzen hinsichtlich des Wasserspiegels, sind numerische Analysen eine wichtige Ergänzung zu einem physischen Modell. Diese Masterarbeit zeigte auch die vielseitigen Möglichkeiten von open-source Software.

Acknowledgements

This thesis was written during my time at the Institute of Hydraulic Engineering and Water Resources Management - Graz University of Technology. Without the guidance and encouragement of several people this thesis wouldn't exist:

First I would like thank to Shervin Shahriari for his great help during this thesis. He introduced me to the huge and complex topic of computational fluid dynamics and proposed this thesis. He provided the hardware and the knowledge that was necessary. His continuous help and dedication assured progress and the finalization of this demanding thesis.

Furthermore I would like to thank all of my friends that had a open ear for my problems regarding this thesis but also for personal problems within this rough time. They helped me laugh and keeping a cool head even in turbulent times.

Last and certainly not least I would like to thank my family, especially my uncle Hubert, my parents Blasius and Cäcilia, and my brother Peter. My uncle I would like to thank for his financial support all over my studies. Without his generosity the whole process of studying would have slowed down. My parents I would like to thank for the never ending support they gave me throughout my academic studies. With their life experience they helped me through all of my small and huge troubles. Special thanks goes to my brother. He helped me out not just once with his own expertise. Despite having stressful times himself he was always there when support was needed.

Statutory declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

November 5, 2017

. . . . . .

Date

. . . . . .

Signature

Contents

Nomenclature

Abbreviations

BC	Boundary Condition
BD	Blended Differencing
BHQ	Design Flood Event
CD	Central Differencing
CFD	Computational Fluid Dynamics
FVM	Finite Volume Method
IWB	Institute of Hydraulic Engineering and Water Resources Management of Graz University of Technology
LUD	Linear Upwind Differencing
PDE	Partial Differential Equation
SHQ	Extreme Flood Event
STL	Stereolithography
UD	Upwind Differencing
VOF	Volume of Fluid method

Constants

κ	Karman's constant	
B	Additive constant for the log-law	
g	Gravity	$\frac{m}{s^2}$

Greek

α	Phase fraction	
ϵ	Turbulent dissipation rate	$\frac{m^2}{s^3}$

Γ	Diffusion coefficient	$\frac{m^2}{s}$
μ	Dynamic viscosity	$\frac{kg}{ms}$
μ_w	Overflow coefficient	
ν	Kinematic viscosity	$\frac{m^2}{s}$
Φ	Face flux field	$\frac{m^3}{s}$
ρ	Density	$\frac{kg}{m^3}$
σ	Surface tension	$\frac{N}{m}$
τ	Shear stress	Pa

Operators

$\nabla\cdot$	Divergence, $Example : \nabla \cdot \mathbf{F} = \sum_{i=1}^{n} \frac{\partial}{\partial x_i} \mathbf{F}^i$	
∇	Gradient, $Example : \nabla(f) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$	

Variables

Δt	Time step	s
$\mathbf{u}$	Velocity vector containing u, v and w	$\frac{m}{s}$
A	Cross section of channel-flow	m^2
C	Courant number	
D_H	Hydraulic diameter	m
H	Water level in the reservoir	$m.a.s.l.$
h_0	Weir overflow height	m
H_{Inlet}	Water level at the models inlet	m
h_{min}	Energy height of approach flow	m
I	Turbulence intensity	
K	Mean curvature of surface	
k	Turbulent kinetic energy	$\frac{m^2}{s^2}$
k_s	Equivalent sand grain roughness	mm

L	Characteristic turbulent length of model	m
l	Turbulent length scale	m
P	Wetted area	m
p	Pressure	Pa
Q	Discharge	$\frac{m^3}{s}$
r	Ratio of upwind to downwind side gradient	
u, v, w	Directional velocities in x, y and z-direction	$\frac{m}{s}$
u^+	Dimensionless velocity	
$u_\star$	Friction velocity	$\frac{m}{s}$
v_0	Approach flow velocity	$\frac{m}{s}$
W	Total width of the Weir	m
y^+	Dimensionless distance from wall boundaries	

1 Introduction

1.1 Introduction

This master's thesis deals with numerical simulations regarding the flow behaviour of a weir with a fixed crest. The numerical models are based on a physical model experiment that was commissioned by Vorarlberger Kraftwerke AG [2013] and carried out at the Institute of Hydraulic Engineering and Water Resources Management (IWB) of Graz University of Technology. The outcome of this model test is presented in a technical report (IWB [2014]). This hydraulic model test should prove the required capacity of an existing weir structure after renovation. The main part of the renovation is a removal of a mid pillar, thus merging the two weir fields. In the physical model test several different weir geometries and a removal of a still existing diversion dam were examined. The results of this physical model test are compared with results acquired from numerical analyses. Furthermore, the performed simulations allow additional results, like pressure distributions at the weir structure, not provided by the model test.

Computational Fluid Dynamics (CFD) analyses of weir overflows and spillways with different setups have been a research field for the last years, starting with more simple models such as in Ho et al. [2001]. With decreasing costs and increasing computational power, the complexity of numerical models increased over the years. Due to this development, CFD became a common tool for engineers to tackle hydraulic problems.

Nowadays there are several different software packages and codes available, which are capable of flow analysis. These codes are distinguishable by the equations which are solved. Furthermore these codes can be categorized whether they are commercial or for free. A former Master's Thesis of IWB to mention is Heinzle [2014]. It also deals with numerical analyses of a weir, conducted with the open-source program *Telemac-3D*, in combination with results of a physical model test. Another common open-source package for fluid dynamics is *OpenFOAM*, which is used in this thesis. Several publications in hydraulic engineering are based on computations with *OpenFOAM* and also showed that the performance is the same compared to commercial tools. Noteworthy is Politano et al. [2016], as they compare computations performed with *ANSYS Fluent* and *OpenFOAM* at weir overflows.

1.2 Objective

The main objective of this thesis is to perform 2D and 3D CFD simulations and to compare acquired results with the related results from the physical model test. The main focus lies on the overflow coefficient μ_w calculated with the Poleni formula. This value is of great interest for evaluating the efficiency of a weir. Using this coefficient allows a comparison of numerical simulations with the physical model test and the current situation. Furthermore, it gives the opportunity to compare the calculated weir efficiency with reference values from literatures. In addition, pressures along the weir are analysed. The last computational goal is an estimation of the required aeration. Since *OpenFOAM* is an open-source software, it is of interest to achieve these objectives only with open-source software. Furthermore, it is important to find an efficient approach for such models.

1.3 Outline

For a general understanding of the examined situation, it is essential to provide some information about the existing weir structure, the physical model test, and its results. This general information is available in Chapter 2. Furthermore, information about the theoretical background regarding CFD is given within Chapter 3. As CFD is a wide topic, only some basic knowledge which is related to this thesis is provided.

The numerical simulations are divided into two parts. First, 2D computations were performed to obtain an overview of the systems sensitivity regarding several different parameters. Information on these simulations is provided in Chapter 4. Furthermore, first comparisons of a numerical analysis to the model are possible with these results. As the investigated weir most likely has a three dimensional influence, 3D-models were computed and are described in Chapter 5. The 3D-computations are based on the findings of the 2D-analyses, this verifies the influence of some parameters. With the acquired data from this analyses, a final comparison of the physical model and the operator's desired state is possible.

In the final chapter, the essential output from both computational chapters is summarized, concluded, and discussed.

2 Physical Model / Weir Gstins

In 2013, a commission for a weir study with a scaled physical model test was given from a power plant's operator to the Institute of Hydraulic Engineering and Water Resources Management (IWB) of Graz University of Technology. This chapter describes the power plant and the weir itself shortly and presents the results achieved from the model test. The information included is mainly from the operator's Vorarlberger Kraftwerke AG [2013] model request and the final assessment from IWB [2014].

2.1 Power Plant "Unterstufe Lutz"

2.1.1 Power Plant

The runoff river power plant Lutz "Unterstufe" was constructed between 1957 and 1959 and is operated by Vorarlberger Kraftwerke AG. It is situated in Bludenz (Vorarlberg) at the river Lutz, which is the main river of the "Großes Walsertal". The power plant is directly connected to the river Ill through a channel.

The main technical data of the power plant can be found on the homepage of the operator Vorarlberger Kraftwerke AG [2017] and is summarized in Table 2.1. A sketch of the power plant's longitudinal cross section, based on information from VORARLBERG ONLINE [2008], is given in Figure 2.1.

Table 2.1: *Technical data - Unterstufe Lutz*

Machinery	Two vertical Francis turbines
Gross head	72 m
Shortage performance turbining	8.6 MW
Number of revolutions	600 rot/min
Generator - nominal voltage	6.4 kV
Generator - nominal performance	6 MVA per generator
Initial operation	1959

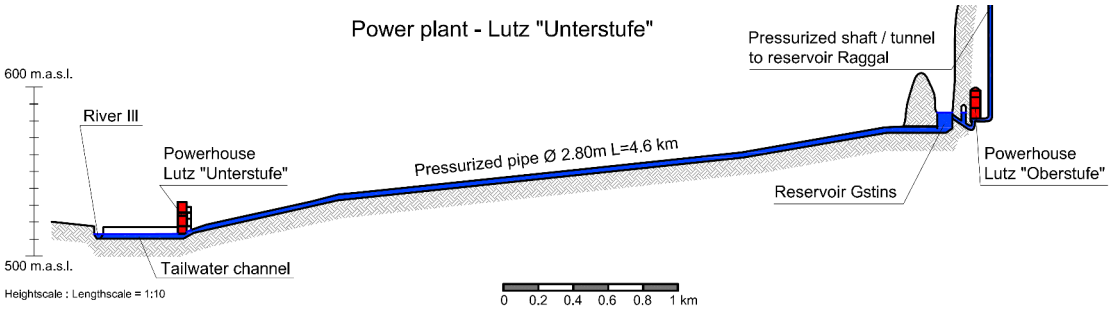


Figure 2.1: *Longitudinal cross section of the power plant Unterstufe Lutz*

2.1.2 Dam / Weir

The dam is located at kilometer 5.15 of the river Lutz. It is constructed as a gravity wall with two flaps. Each of those weir fields has a width of 10 meters and they are separated with a pillar. The intake is situated on the orographic right side of the river about 60 upstream of the dam. The intake leads to a 4.6 kilometre long pressurized tunnel. An overview of the site and a photograph of the existing weir are given in Figures 2.2 and 2.3. Data regarding reservoir and the current flow over the spillway are listed in Table 2.2.

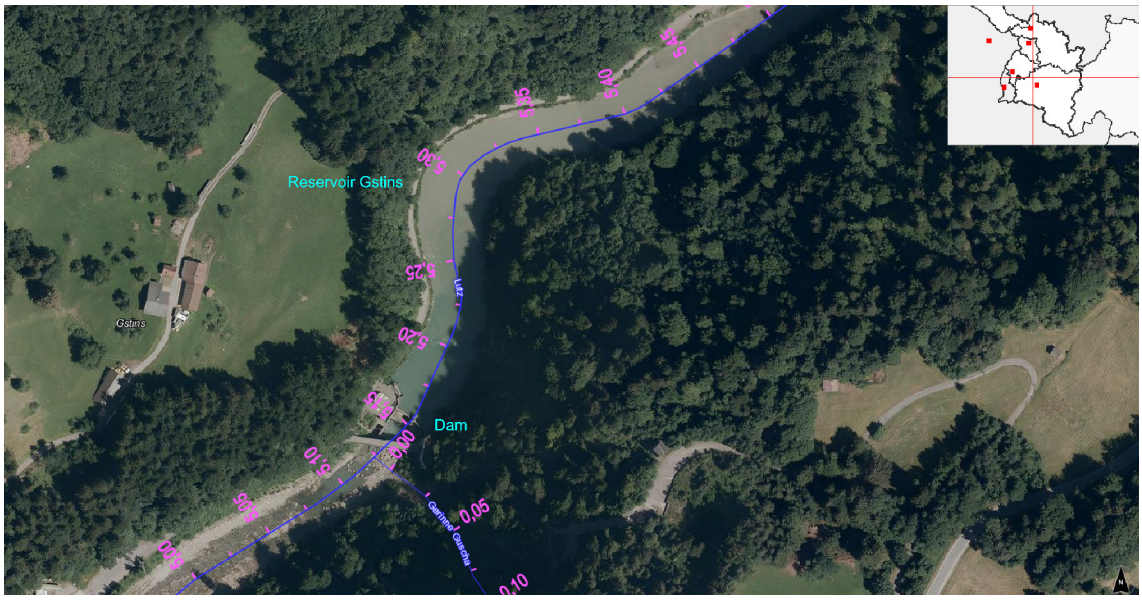


Figure 2.2: *Project site - VoGIS [2017]*

Flood events in the past years have shown that there is a problem with driftwood jam. Therefore, it is planned to remove the pillar and thus replace the two 10 meter weir fields by a single 20 meter field. According to a 5 year inspection, also the design discharges have changed. These values are as follows:

- BHQ = 397 m^3/s
- SHQ = 679 m^3/s

Table 2.2: *Technical data - Reservoir Gstins - Vorarlberger Kraftwerke AG [2013]*

Catchment area	$\approx$ 180 km^2
Retention water level	585.0 m.a.s.l.
Minimum operating level	582 m.a.s.l.
Reservoir length	$\approx$ 500 m
Reservoir volume	$\approx$ 50,000 m^3
Current design flood event (80 cm flood surcharge with opened bottom outlet)	475 m^3/s
Discharge with lowered flaps and retention water level (585.0 m.a.s.l.)	296 m^3/s
Discharge with lowered flaps and flood surcharge level (585.80 m.a.s.l.)	383 m^3/s

The suggested alterations at the weir are given in Figures 2.4 and 2.5. Larger scale plans and more photographs of the current situation are included in Appendix 8.2. As shown, the pillar will be removed and the side pillars will be slightly extended. The weir itself will get broader and the alignment in respect to the reservoir will change. Due to a weir axis rotation, the flow will have no perpendicular direction to the spillway. Also shown in Figure 2.4 is a diversion dam that is still existing in the reservoir.

Figure 2.3: *Photograph weir - 23.12.2012 - Vorarlberger Kraftwerke AG [2013]*

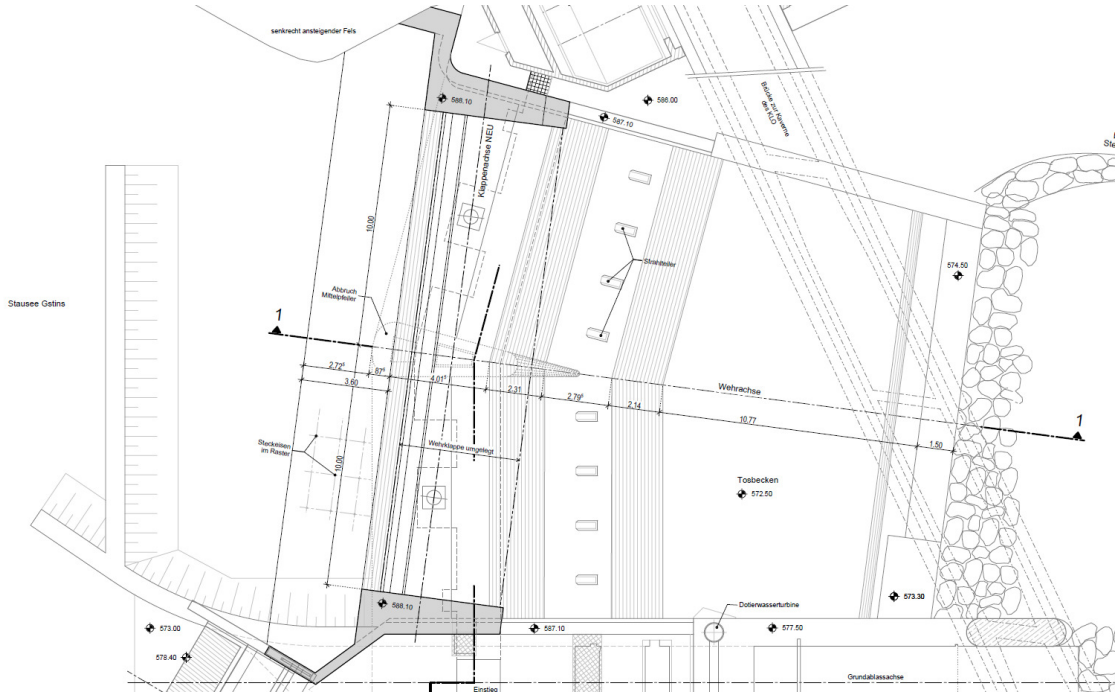


Figure 2.4: *Alteration weir - Top view - Vorarlberger Kraftwerke AG [2013]*

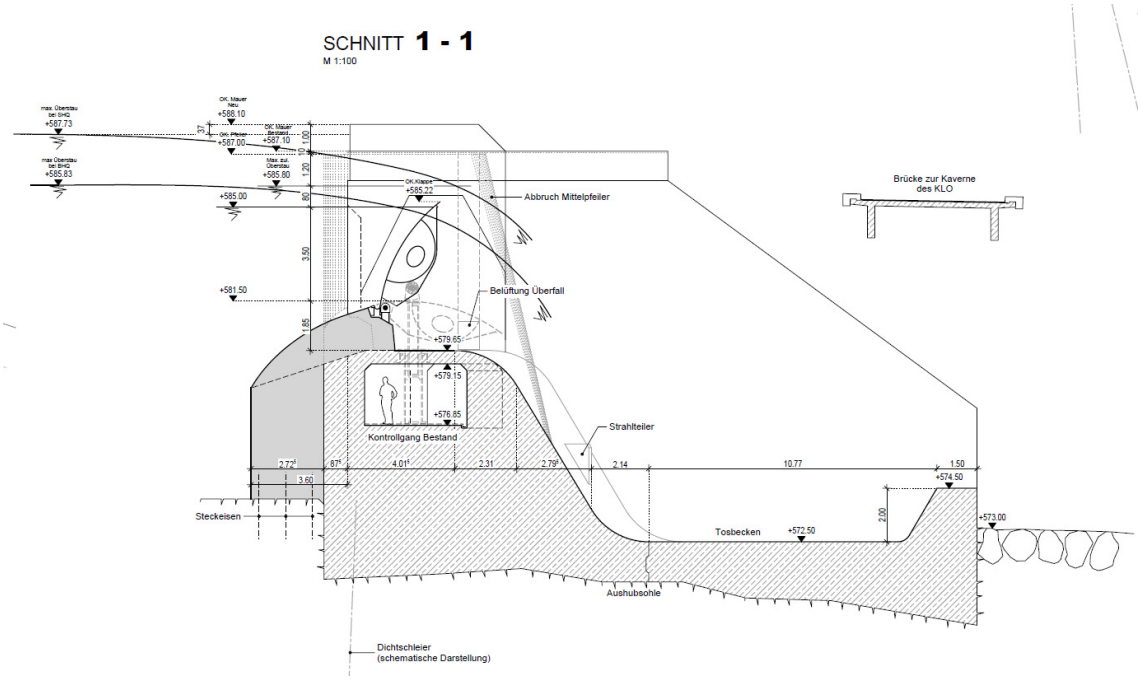


Figure 2.5: *Alteration weir - Cross section - Vorarlberger Kraftwerke AG [2013]*

2.2 Physical Model Test

The physical model test was carried out in a glass flume in the laboratory of IWB. It was done with a 1:30 scale by use of the Froude-similarity.

2.2.1 Objective

The main objectives of the physical model test are (Vorarlberger Kraftwerke AG [2013]):

- Verification of the weir's discharge by use of common hydraulic calculations. Creation of a flow/height diagram up to a maximum flow of SHQ.
- Verification of the weir's discharge by use of a cutaway model for BHQ and SHQ. Maximum flow should be approximately 750 m^3/s (SHQ +10%)
- Verification of the maximum water levels in the reservoir for BHQ and SHQ. No further discharges (bottom outlet, operation of power plant) should be considered.
- Investigation of the energy dissipation in the existing stilling basin for BHQ and SHQ by use of common literature. Estimation of necessary changes.
- Verification of the energy dissipation for the existing and in case changed spillway by use of a cutaway model with a maximum flow of the SHQ.
- Determination of protection measurements after the stilling basin for BHQ and SHQ. Regarding the river bed there are two main principles:
  - BHQ: No damages at the river bed
  - SHQ: Small damages acceptable, but no risk to the stability of the bed
- Optional: Computational investigation of the inflow consistency with a hydraulic program.

2.2.2 Model Setup

The model setup was done within a flume with simplified geometry. The flume had a height of one metre, a width of 0.67, and a length of approximately 10 metres. The basic layout of the model within the flume is shown in Figures 2.6 and 2.7. More detailed plans of the setup are shown in Appendix 8.3.

In addition to the initial state defined by the operator, several different versions of weir and stilling basin were tested. These variations included two weir profiles that are shown in Figure 2.8 and seven versions of the stilling basin. First trials indicated an unrealistic contraction, especially at the right side, which is not showing in nature. This happened most likely because of the sharp form of the side pillars and the missing enlargement upstream due to the reservoir. Therefore, two new extensions to the side walls were introduced in the model shown in Figure 2.9 to achieve a more natural state with the model.

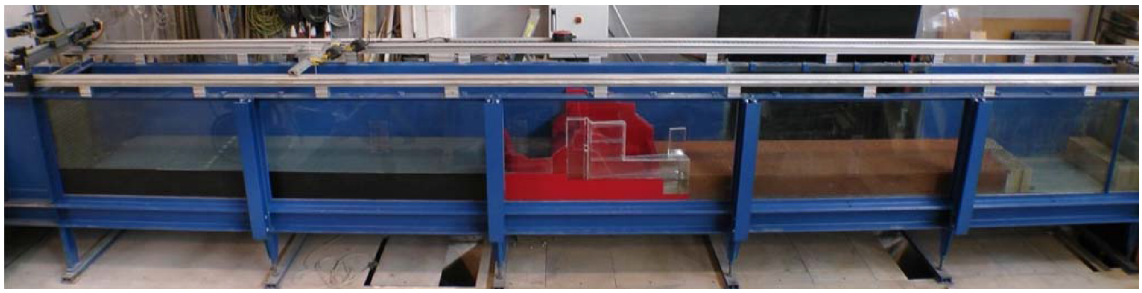


Figure 2.6: *Overview flume - Model scale 1:30 - IWB [2014]*

Figure 2.7: *Overview model - Initial state - Model scale 1:30 - IWB [2014]*

As shown in Figure 2.10, the measurement of water levels for comparison with the existing state was done 45 metres upstream of the flaps rotation axis. The basic material used for the model itself and the flumes bottom was Trovidur PVC. The right side pillar of the model was created with acrylic glass. The left side of the channel was made out of coated steel, the right one out of glass. Later during the tests, fine grained terracotta material with diameters from 3 to 8 millimetres was added to the downstream part of the model.

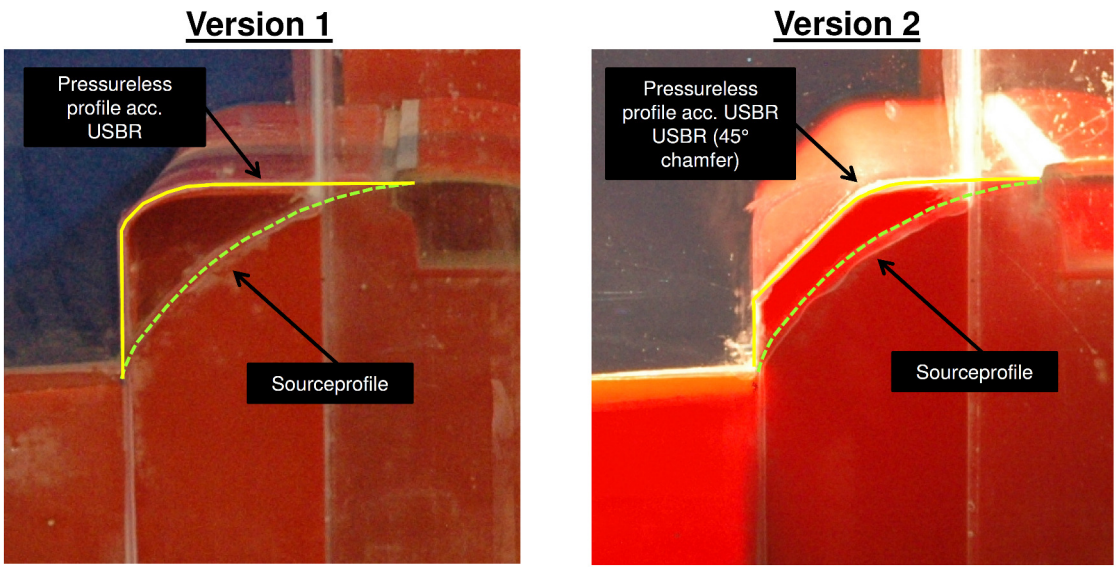


Figure 2.8: *Weir profiles - IWB [2014]*

(a) Pillar form 1 - Linear extension (b) Pillar form 2 - Elliptic extension

Figure 2.9: *Pillar extensions - IWB [2014]*

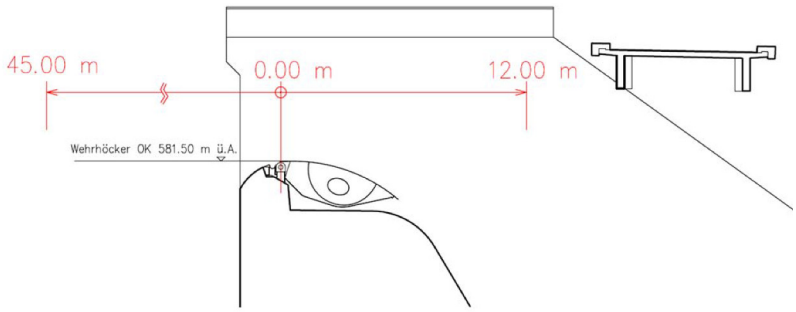


Figure 2.10: *Location - Point of measurement - IWB [2014]*

2.2.3 Results

Seven main results were used for comparison of the weir overflow. These contain the desired value, which is the current state with two weir fields specified by the operator, and six different test in the physical model with altered weir and side pillar forms and a removed construction diversion dam.

The most common way to compare the efficiency of different versions is the overflow coefficient μ_w, which is used in the Poleni and Weissbach formulas. In the original report the upstream velocity was neglected and only the Poleni formula in Equation 2.1 was used. This formula can be used to recalculate μ_w with the weir's width W and the overflow height h_0. The second possibility would have been the usage of the Darcy-Weisbach Equation 2.2. This formula uses the same overflow coefficient but takes the approach flow velocity v_0 entering the weir's area into account. This velocity can be calculated with simple continuity. A common criteria for the usage of the Weisbach formula is a velocity $v_0 > 1.5m/s$, as below it has no significant influence. A sketch of all depending variables for these formulas is provided in Figure 2.11

$$Q_{Poleni} = \frac{2}{3} \cdot \mu_w \cdot W \cdot \sqrt{2g} \cdot h_0^{\frac{3}{2}} \tag{2.1}$$

$$Q_{Weisbach} = \frac{2}{3} \cdot \mu_w \cdot W \cdot \sqrt{2g} \cdot \left[\left(h_0 + \frac{v_0^2}{2g} \right)^{\frac{3}{2}} - \left(\frac{v_0^2}{2g} \right)^{\frac{3}{2}} \right] \tag{2.2}$$

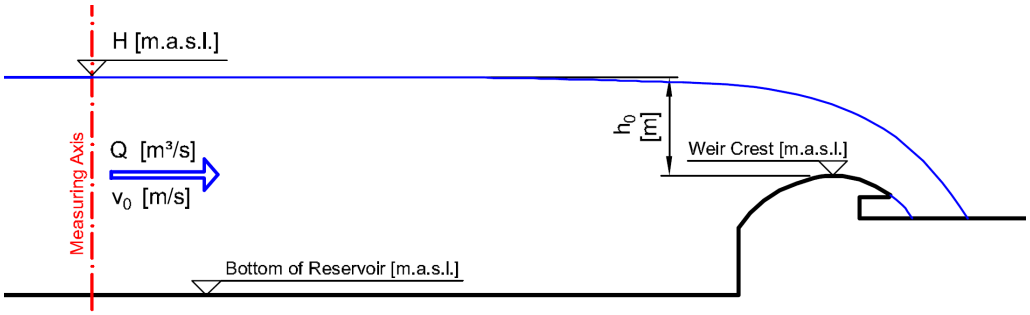


Figure 2.11: *Sketch of depending variables*

Depending on the weir's shape, there are several μ_w values available from literature. The coefficient for comparison was chosen for a fully rounded, broad weir (e.g. fish belly flap fully opened) from Rössert [1994].

$$\mu_w \approx 0.65 - 0.73 \tag{2.3}$$

All seven results are compared with their water levels and discharge coefficients calculated with Poleni formula for BHQ and SHQ in Table 2.3. As can be seen in this table, all values of the model fall below the current natural state, decreasing the weir's efficiency by at least 8% and up to 22%. Compared to the literature's values from Equation 2.3, only the ones with the modified pillar were within this range. The usage of the Weisbach equation instead of Poleni would lead to a further decrease of their respective overflow coefficient. The best performance was reached with the initial weir shape, an elliptic pillar modification, and without the diversion dam. Therefore, this model was used for final comparison. The results of this model are highlighted in Table 2.3.

Table 2.3: *Comparison - Weir overflow physical model test*

Setup	$\mathbf{H_{BHQ}}$ $[\mathbf{m.a.s.l.}]$	$\boldsymbol{\mu}_{\mathbf{w,BHQ}}$ $[-]$	$\mathbf{H_{SHQ}}$ $[\mathbf{m.a.s.l.}]$	$\boldsymbol{\mu}_{\mathbf{w,SHQ}}$ $[-]$
Vorarlberger Illwerke - Desired values	585.90	0.73	587.73	0.74
TU Graz				
Initial state	586.50	0.60	588.39	0.64
Weir version 1	586.69	0.57	588.64	0.60
Weir version 2	586.60	0.58	588.55	0.61
Mod. side pillar 1 (Initial weir)	586.41	0.62	588.27	0.65
Mod. side pillar 2 (Initial weir)	586.33	0.63	588.19	0.66
Without diversion dam (Initial weir / Pillar form 2)	586.23	0.65	588.07	0.68

A comparison of all variants regarding their water level is given in Figure 2.12. It shows an increase of the water level for both versions of the weir's shape and a decrease of the water level with modified side pillars. Recall that the pillar modifications were only used for decreasing contraction in the model which is not occurring in nature. The construction diversion dam had a large impact on the flow situation at the weir's right side. The positive effect of a diversion dam removal was significant. Photographs of the model under operation, which show the effect of dam and side pillar modification, are shown in the Appendix 8.3.2.

A final comparison of the required water heights and flows to the model without diversion dam is given in Tables 2.4 and 2.5. The first one compares the required water level with their maximum flow in the model, the second one the required discharge with the reservoirs elevation. These tables show that a modification of the weir decreases the possible discharge. As the existing structures top has a elevation of 588.10 m.a.s.l., a freeboard of only 3 centimetres would remain in the SHQ case.

Table 2.4: *Comparison - Difference in discharge of model and desired value*

Desired water level Vorarlberger Illwerke	**Discharge (Model - Desired Value)**	**Deviation to desired value**
585.90 m.a.s.l.	348 m^3/s (Desired capacity = 397 m^3/s)	-49 m^3/s ($\approx$ -12 %)
587.73 m.a.s.l.	634 m^3/s (Desired capacity = 679 m^3/s)	-45 m^3/s ($\approx$ -7 %)

Table 2.5: *Comparison - Difference in water level of model and desired value*

Discharge	**Water level (Model - Desired value)**	**Deviation to desired value**
BHQ = 397 m^3/s	586.23 m.a.s.l. (Desired water level = 585.90 m.a.s.l.)	+ 0.33 m
SHQ = 679 m^3/s	588.07 m.a.s.l. (Desired water level = 587.73 m.a.s.l])	+ 0.34 m

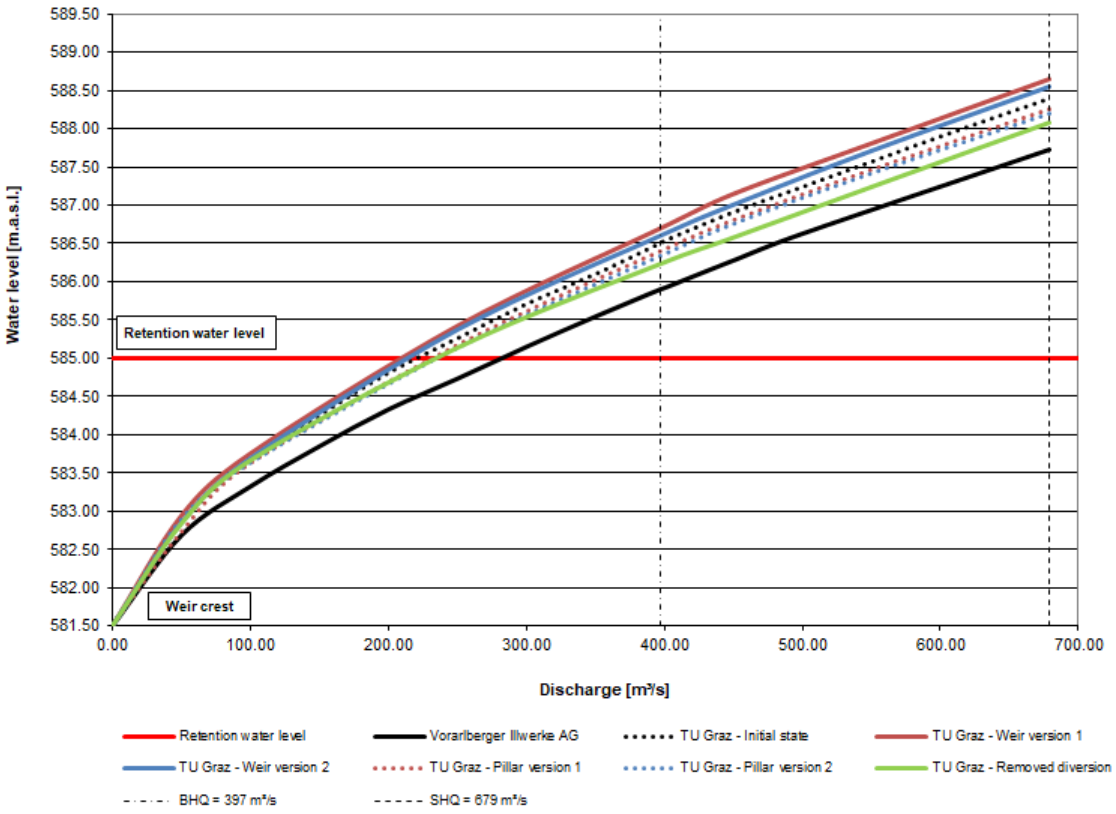


Figure 2.12: *Comparison - Weir overflow - IWB [2014]*

The final report also states a decrease of the aeration area due to the pillar removal. The existing structures aeration area will decrease from 2.16 m^2 to 1.25 m^2. In the physical model a filling beneath the flap was observed for SHQ shown in Figures 2.13 and 2.14, implying insufficient aeration. A validation of the minimum area was performed with a formula from Strobl and Zunic [2006], which is given in Equation 2.4. This formula estimates the necessary area for ventilation by use of the weir's width W and the height difference of the lowered flap to the downstream water level z.

$$A = 0.015 \cdot W \cdot z \tag{2.4}$$

With a width of 20 metres, a maximum flap level of 585.22 m.a.s.l. and a downstream water level of 578.50 m.a.s.l. this led to a approximated aeration area of 2 m^2 for SHQ.

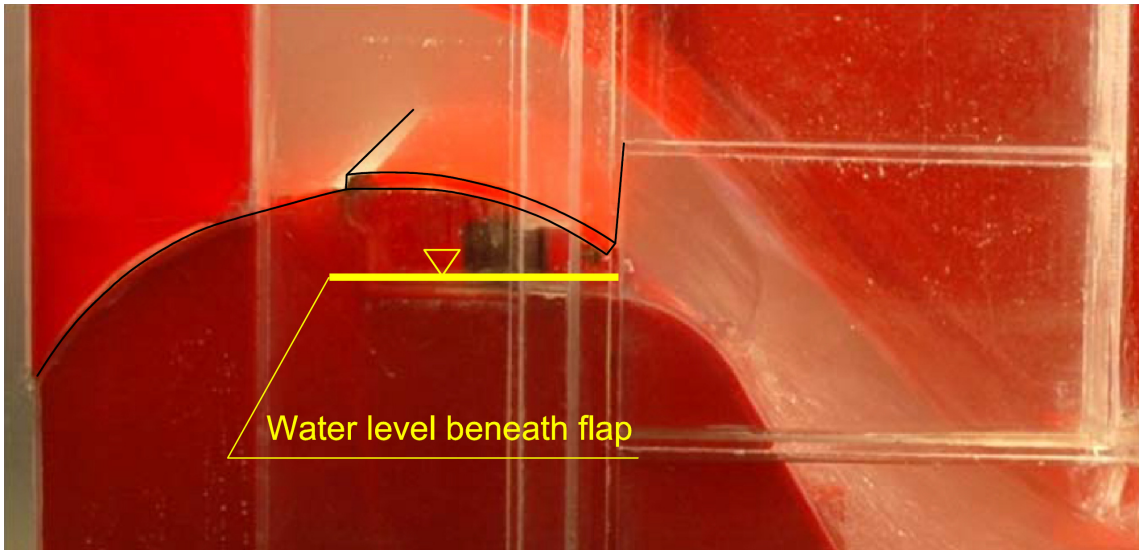


Figure 2.13: *Water level beneath of the flap - BHQ - IWB [2014]*

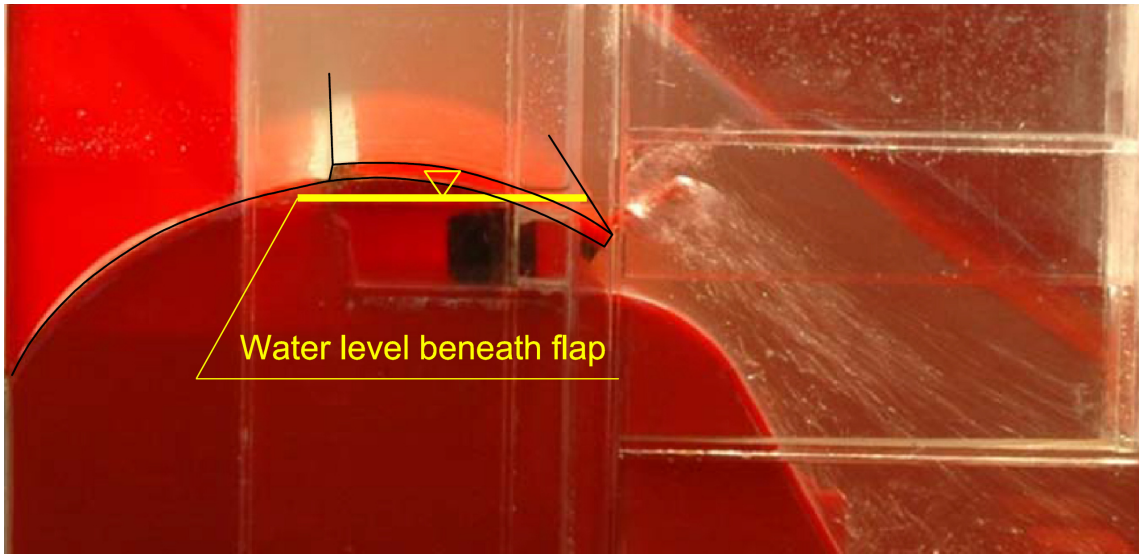


Figure 2.14: *Water level beneath of the flap - SHQ - IWB [2014]*

3 Preliminaries

In this chapter some background needed for understanding this work is given. First, the basic principles of Computational Fluid Dynamics and multiphase analysis are described briefly. Then, information regarding divergence schemes, turbulence modelling, and boundary layers is provided. At last, the snappyHexMesh algorithm that is implemented in *OpenFOAM*, is described.

3.1 CFD / OpenFOAM

Computational Fluid Dynamics is a method to solve tasks that involve fluid flows numerically. The implementation of the numerical model was done with the open-source software *OpenFOAM*. It is a program designed for CFD by use of the finite volume method. This powerful tool can be used for simulations in various different fields and it is possible to modify it.

3.1.1 CFD Basics

The basic equations of the finite volume method (FVM) for fluid dynamics are given in this section. More detailed information can be found in Versteeg and Malalasekera [2007].

The governing equations of fluid flow are continuity, momentum, and energy. These are based on the conservation laws of physics and the principle of a Newtonian fluid. Each of those leads to a set of partial differential equations (PDE) to solve. Those laws can be summed up as:

- The mass of a fluid is conserved $\Rightarrow$ Continuity equation
- The rate of change of momentum equals the sum of the forces on a fluid particle (Newtons second law) $\Rightarrow$ Momentum equation
- The rate of change of energy is equal to the sum of the rate of heat addition to and the rate of work done on a fluid particle (first law of thermodynamics) $\Rightarrow$ Energy equation

As weir overflows have a relatively low velocity the fluid can be seen as incompressible. Therefore, the density ρ remains constant which leads to a decoupling of the energy equation. As no heat transfer is considered in this thesis, the system could be solved

by mass conservation and momentum equations only. Thus more detailed information about the energy equation are not given here but can be found in Versteeg and Malalasekera [2007]. The directional velocities for x,y, and z can be summed up within a vector $\mathbf{u} = [u, v, w] = [v_x, v_x, v_z]$.

An example used for deriving the continuity of a finite volume element is given in Figure 3.1. The situation that can be seen leads to the continuity equation for a three dimensional, unsteady, and compressible flow. It is given in directional and short vector notation in Equation 3.1. Note that the first term $\frac{\partial \rho}{\partial t}$ can be neglected in an incompressible case as the density is constant.

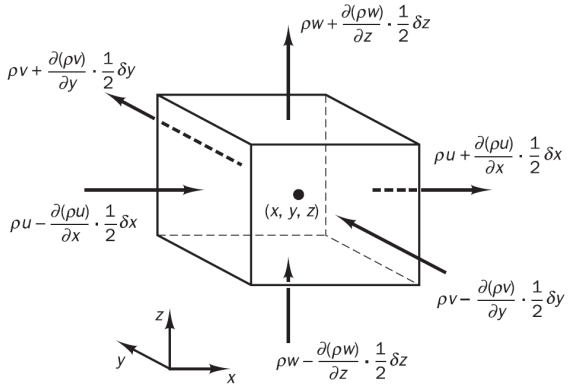
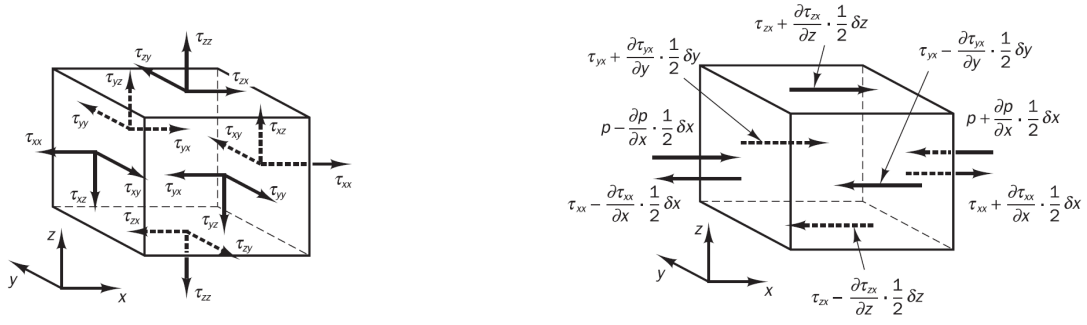


Figure 3.1: *Mass flow within a fluid element - Versteeg and Malalasekera [2007]*

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = \frac{\partial \rho}{\partial t} + div(\rho \mathbf{u}) = 0 \tag{3.1}$$

In general, one can distinguish between surface and body forces acting on a fluid element. The momentum equation contains the surface forces. These are pressure and viscous stresses and are represented in Figure 3.2, which shows the general stress components on the left side and in more detail only in x-direction on the right side. Summing up the forces shown in this figure leads to the momentum Equation 3.2 for the x-direction, which gives the increase of x-momentum on a particle. The source term S_M collects all outside (body) forces. An easy example could be given for gravity in negative z direction with $S_{Mz} = -\rho g$.

$$\rho \frac{Du}{Dt} = \frac{\partial(-p + \tau_{xx})}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + S_{Mx} \tag{3.2}$$



(a) Stress components on fluid element (b) Stress components in x-direction

Figure 3.2: *Stress components on a fluid particle - Versteeg and Malalasekera [2007]*

Under the assumption of a Newtonian fluid in which the viscous stress are proportional to the rate of deformation, the momentum equations lead to the so called Navier-Stokes equations given in Equations 3.3, 3.4, and 3.5.

$$\rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + div(\mu \; grad \; u) + S_{Mx} \tag{3.3}$$

$$\rho \frac{Dv}{Dt} = -\frac{\partial p}{\partial y} + div(\mu \; grad \; v) + S_{My} \tag{3.4}$$

$$\rho \frac{Dw}{Dt} = -\frac{\partial p}{\partial z} + div(\mu \; grad \; w) + S_{Mz} \tag{3.5}$$

Summarizing those formulas leads to the governing equations of a time dependent three dimensional fluid flow for a compressible state. The continuity equation is given in 3.6 and momentum (x,y,z) in equations 3.7, 3.8, and 3.9. Together they build a set of partial differential equations of non-linear second order. These are only solvable for finite difference (FDM), element (FEM), and volume (FVM) method. For CFD, only FVM is of interest.

$$\frac{\partial \rho}{\partial t} + div(\rho \mathbf{u}) = 0 \tag{3.6}$$

$$\frac{\partial \rho u}{\partial t} + div(\rho u \mathbf{u}) = -\frac{\partial p}{\partial x} + div(\mu \; grad \; u) + S_{Mx} \tag{3.7}$$

$$\frac{\partial \rho v}{\partial t} + div(\rho v \mathbf{u}) = -\frac{\partial p}{\partial y} + div(\mu \; grad \; v) + S_{My} \tag{3.8}$$

$$\frac{\partial \rho w}{\partial t} + div(\rho w \mathbf{u}) = -\frac{\partial p}{\partial z} + div(\mu \; grad \; w) + S_{Mz} \tag{3.9}$$

The terms of the momentum equation on the example of the x-direction are:

- Unsteady acceleration: $\frac{\partial \rho u}{\partial t}$
- Convective acceleration: $div(\rho u \mathbf{u})$
- Pressure gradient: $\frac{\partial p}{\partial x}$
- Diffusion: $div(\mu \; grad \; u)$
- Source term: S_{Mx}

Recall that this is the set of terms is valid for a compressible, unsteady fluid flow. As mentioned before, the term $\frac{\partial \rho}{\partial t}$ gets 0 in an incompressible case. Furthermore, the unsteady acceleration can be neglected in a steady state flow.

All those equations have a very similar form, therefore it is possible to express those with a general formula like in Equation 3.10 using the flow variable Φ. This is the most common version of the transport equation. A full derivation of this formula starting from basic continuum mechanics is also available from Jasak [1996].

$$\frac{\partial \rho \Phi}{\partial t} + div(\rho \Phi \mathbf{u}) = div(\Gamma \; grad \; \Phi) + S_\Phi \tag{3.10}$$

This equation could be expressed in words like in Versteeg and Malalasekera [2007].

Rate of increase of ϕ of fluid element	+	Net rate of flow of ϕ out of fluid element	=	Rate of increase of ϕ due to diffusion	+	Rate of increase of ϕ due to sources

3.1.2 Multiphase analysis

In general there are two types of multiphase methods available for *OpenFOAM*. They can be separated as in Schulze and Thorenz [2014] by their basic principle as follows:

- Volume-of-Fluid method (appropriate for clearly separated phases)
- Eulerian-Eulerian approach (used for disperse phases)

For this thesis the Volume of Fluid method (VOF) was of importance. It was introduced by Hirt and Nichols [1981]. This means that for each cell an additional advection equation has to be solved in addition to the Navier-Stokes equations. With this step it is possible to determine a volume fraction between two liquids (e.g. water and air) for each cell. There are three possible states for the volume fraction.

- $\alpha = 1.0 \rightarrow$ Cell completely filled with water
- $\alpha = 0.0 \rightarrow$ Cell completely filled with air
- $1.0 > \alpha > 0.0 \rightarrow$ Interface cells

In the interface cells density and viscosity of liquid and gas are weighted according to their phase fraction. The weighing of those variables is carried out as in Equations 3.11 and 3.12. The volume fraction is visualised in Figure 3.3. It also already suggests that a constant cell level at the main intersection area of water and air is of advantage.

$$\rho = \rho_l \alpha + \rho_g (1 - \alpha) \tag{3.11}$$

$$\rho = \mu_l \alpha + \mu_g (1 - \alpha) \tag{3.12}$$

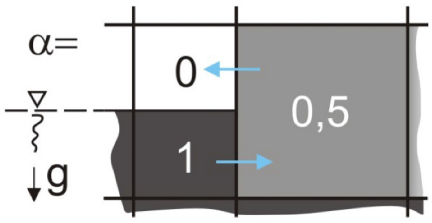


Figure 3.3: *Example volume fraction α - Gisen [2014]*

There are two ways to find the actual water surface, either by tracking an alpha value of 0.5 with interpolation of the cells values, or by reading pressures at the reservoirs ground. The pressure existing at the bottom can be seen as the pure hydrostatic pressure, so the second possibility are pressure probes along the ground.

In this thesis only the `interFoam` multiphase solver of *OpenFOAM* was used as it is one of the most common solvers for such cases. The final equations to solve are continuity, phase fraction, and the modified multiphase version of the momentum equation. Those

are given in Equations 3.13, 3.14, and 3.15 like in Damian [2012]. These equations are already rearranged and discretized.

$$\nabla \cdot \mathbf{u} = 0 \tag{3.13}$$

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\mathbf{u}\alpha) + \nabla \cdot [\mathbf{u}_r \alpha(1-\alpha)] = 0 \tag{3.14}$$

$$\frac{\rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla \cdot (\mu \nabla \mathbf{u}) - (\nabla \mathbf{u}) \cdot \nabla \alpha = -\nabla p_d - \mathbf{g} \cdot \mathbf{x} \nabla \rho + \sigma K \nabla \alpha \tag{3.15}$$

In the modified version of the phase fraction, the relative velocity vector $\mathbf{u}_r$ is used, which can be calculated with the velocities of water and air like in Equation 3.16. Compared to the momentum Equations from 3.7 to 3.9, the diffusion term in Equation 3.15, based on the deviatoric stress tensor, is decomposed and brought to the right side. Furthermore, the pressure gradient is modified by separation of hydrostatic and hydrodynamic pressure as in Equation 3.17. Due to this, body forces caused by pressure gradient and gravity are included in these terms. The term $\sigma K \nabla \alpha$, which is added to the basic equation, deals with the effect of surface tension at the interface of water and air. This reaction creates an additional pressure gradient, which is based on the mean curvature K of the surface. For a full derivation of these formulas and further information about the solver, see Damian [2012].

$$\mathbf{u}_r = \mathbf{u}_w - \mathbf{u}_a \tag{3.16}$$

$$p = p_d + \rho \mathbf{g} \cdot \mathbf{x} \tag{3.17}$$

3.2 Divergence schemes

For each of the calculation terms there are several different interpolation schemes available in *OpenFOAM*. For this thesis the divergence schemes are of great interest, especially those used for the convection term $\nabla \cdot (\rho \mathbf{u}\mathbf{u})$ or `div(phi,U)` in *OpenFOAM*. The `Gauss` scheme is always used for discretization, but the different interpolation schemes available can be found in the *OpenFOAM* User Guide of Greenshields [2017]. Some of these schemes are listed in Table 3.1. These schemes decide which way is used to interpolate for the face field ϕ_f.

The order of a scheme describes its computational accuracy. The term boundedness describes the behaviour of internal node values. Without any extra sources this means that the nodal values are bounded to their neighbouring values. Furthermore they should all have the same sign. This implies that an increase of ϕ at one node leads to an increase of ϕ at the neighbouring node.

Table 3.1: *Divergence schemes used for convection term*

Scheme	Type	Numerical behaviour
`upwind`	Upwind differencing	1^{st} order, bounded
`linearUpwind`	Linear upwind differencing	1^{st}/2^{nd} order, bounded
`linear`	Linear interpolation (central differencing)	2^{nd} order, unbounded

The implementation of this schemes is shortly described in the *OpenFOAM* Programmer's Guide of Greenshields [2015]. Cell values are noted in capital and faces in small letters. In the following formulas N represents the upstream (upwind) value and P the downstream (downwind) value.

Upwind differencing (UD) determines ϕ_f directly from the neighbouring cell dependent on the flow direction F. This is given in Equation 3.18.

$$\phi_f = \begin{cases} \phi_P \ for \ F \geq 0 \\ \phi_N \ for \ F < 0 \end{cases} \tag{3.18}$$

A special form of upwind biased schemes is the linear upwind differencing (LUD), which is given in Equation 3.19. This method involves two upstream values, leading to a second order accuracy keeping its boundedness. In addition to the upstream value, the face value is calculated with a vector from the cell centre to the face $\overline{Pf}$ and the upstream gradient.

$$\phi_f = \phi_P + \overline{Pf}\nabla\phi_{upstream} \tag{3.19}$$

Central differencing (CD) uses the upstream and the downstream cell to estimate ϕ_f. The implementation of this scheme is given in Equation 3.20. It depends on the weighting factor f_x used for consideration of the distances. This variable is described in Equation 3.21 in which $\overline{fN}$ is the distance from the face f to the cell center N and $\overline{PN}$ the distance between the upstream and downstream cell centres.

$$\phi_f = f_x\phi_P + (1 - f_x)\phi_N \tag{3.20}$$

$$f_x = \frac{\overline{fN}}{\overline{PN}} \tag{3.21}$$

Blended differencing (BD) is a combination of upwind and central differencing schemes,

by use of a blending coefficient γ, given in Equation 3.22. These schemes are used for maintaining the boundedness of UD with a higher accuracy of CD. This might be necessary as CD has convergence problems with sharp gradients of the dependent variables.

$$\phi_f = (1-\gamma)(\phi_f)_{UD} + \gamma(\phi_f)_{CD} \tag{3.22}$$

$$\gamma_{Van\ Leer} = \frac{r+|r|}{1+r} \tag{3.23}$$

There are several different ways to compute the blending coefficient γ. Common methods are already implemented in *OpenFOAM*. One of the most used is that by Van Leer [1974] given in Equation 3.23, in which r represents the ratio of upwind-side to downwind-side gradient. The Van Leer scheme is often used to solve for the phase fraction α in multiphase analysis.

3.3 Turbulence modelling

In this work, a standard k-ϵ turbulence model was used. This is one of the most common turbulence models available for hydraulic purposes. The input parameters for *OpenFOAM* using this turbulence model are the turbulent kinetic energy k and the rate of turbulent dissipation ϵ. Those can be calculated as constants from a chosen turbulence intensity I as in Equations 3.24 - 3.27 from Versteeg and Malalasekera [2007].

$$k = \frac{3}{2}(u_{ref}I)^2 \tag{3.24}$$

$$u_{ref} = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{3.25}$$

$$\epsilon = \frac{0.1643k^{1.5}}{l} \tag{3.26}$$

$$l \approx 0.07L \tag{3.27}$$

L serves as the characteristic length of a model. For open channel flows in 3D it is a common approach to use the hydraulic diameter, for 2D the water height. As the distribution of k and ϵ is not constant over the depth in nature, another possibility is to use experimental data. Such formulas based on experiments for open channel flows are given in equations 3.28 and 3.29 from Nezu [1993]. In these formulas y represents the position in depth, $u_\star$ the friction velocity, and E_1 is a weak function of the Reynolds number. The friction velocity $u_\star$ can be calculated with the wall shear stress τ_w as in Equation 3.30.

$$\frac{k}{u_\star^2} = 4.78 \cdot e^{-2y/h} \tag{3.28}$$

$$\frac{\epsilon h}{u_\star^3} = E_1 \cdot (\frac{y}{h})^{-1/2} \cdot e^{-3y/h} \tag{3.29}$$

$$u_\star = \sqrt{\frac{\tau_w}{\rho}} \tag{3.30}$$

3.4 Boundary layers / Near wall modelling

There are several different ways to deal with walls in CFD. A common approach for the boundary along walls is the `noSlip` condition. The velocity at such walls is zero, therefore the influenced variables have a very steep gradient. Common meshes are not capable of solving this area as the first node is too far away from the wall. This is represented in Figure 3.4. Therefore, wall functions have been introduced. Wall functions are in general used for the turbulence properties k and ϵ and furthermore for the turbulent viscosity ν_t.

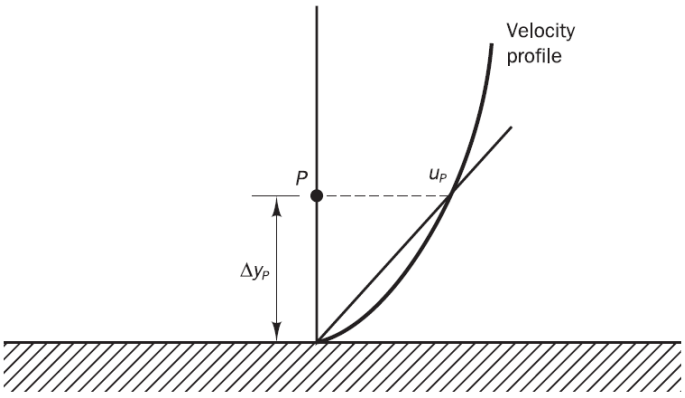


Figure 3.4: *Velocity profile along walls - Versteeg and Malalasekera [2007]*

The near wall area is subdivided in different regions, namely the viscous sublayer, the buffer layer, and the log-law region. The most important variable that defines in which area the first point of the mesh lies in is the dimensionless distance y^+, which is calculated as in equation 3.31.

$$y^+ = \frac{\Delta y_p}{\nu} \cdot u_\star \tag{3.31}$$

This factor depends on the distance to the wall boundary. Especially of interest are the distance to the first mesh point represented by Δy_p and the friction velocity $u_\star$, which is calculated with the wall shear stress τ_w as given in Equation 3.30. There are many

different predefined wall functions available for *OpenFOAM*. The functions used within this thesis are as follows.

1. `epsilonWallFunction`
2. `kqRWallFunction`
3. `nutkWallFunction`
4. `nutkRoughWallFunction`

Functions 3 and 4 refer to the turbulent viscosity ν_t, the first one considering a smooth and the second one a rough wall. All of these wall functions are valid for the log law region.

The log law is given in Equation 3.32. It calculates the dimensionless velocity u^+ with y^+ and two constants. These two constants are the Karman's constant κ and the additive constant B. Values for those can be found in literature like Versteeg and Malalasekera [2007]. The log law gives a good fit to a real velocity distribution within a certain range ($30 < y^+ < 500$). The minimum value for the log law is at y^+ = 11.63. To cover locally higher velocities, it is recommended to aim for maximum values of y^+ = 300 within a main field.

$$u^+ = \frac{1}{\kappa} ln(y^+) \tag{3.32}$$

$$u^+ = \frac{u}{u_\star} \tag{3.33}$$

According to Equation 3.33, adding roughness changes the log law significantly. Roughness leads to an increase of the wall shear stresses and therefore the log law is shifted downwards eventually leading to a disappearing viscous sub-layer.

3.5 Meshing with snappyHexMesh

SnappyHexMesh is one of the meshing tools provided by *OpenFOAM*. It is able to create meshes using hexahedral and split-hexahedral elements based on input STL (Stereolithography) files. For faster creation of meshes it has the advantage of parallel processing.

The following steps are carried out during a mesh creation with snappyHexMesh, as described by Greenshields [2017]. They are visualized with examples from the user guide in Figure 3.5.

a) Creation of STL surfaces with CAD software

b) Creation of a bounding mesh using `BlockMesh`

c) Refinement along surfaces or featured edges created using `surfaceFeatureExtract`

d) Removal of unnecessary cells

e) Additional regional refinement

f) Snapping of the surfaces

g) Layer addition

STL surfaces are a common way to transfer CAD-modelled geometry from one program to another. Surfaces created this way consist of tessellated triangular planes. Depending on the way of creating those surfaces it may already be of great importance to choose a proper resolution of the tessellation to match curved surfaces. Furthermore, it is necessary to separate the geometry into multiple files, one for each necessary patch for the boundary conditions defined later on.
The bounding mesh should be created with a face aspect ratio of approximately 1 to avoid later complications. Also, it should be kept in mind which minimum cell size is necessary after refinement. As the refinement of `snappyHexMesh` is defined with levels, the final cell size could be simply calculated using Equation 3.34.

$$Cell\ size = \frac{Base\ size}{2^{Refinement\ level}} \tag{3.34}$$

Refinements can be defined in several ways. One possibility is to define a range of refinement levels for an input surface. The level chosen by the meshing algorithm can be regulated with the refinement controls. The number of refinement cells created for each level can be manipulated with `nBufferCells`. Another possibility is to define a thickness of fixed refinement layers using predefined featured edges. The last way of refinement is regional refinement independent from surfaces. This, for example could be of advantage for multiphase analysis at the interface between two fluids.

The last step of the meshing process is the layer addition, which first shrinks the mesh to create parallel layers along surfaces or featured edges. The thickness of added layers is defined with a ratio of the neighbouring cell size. This is of interest for solving the boundary layers described in Section 3.4.

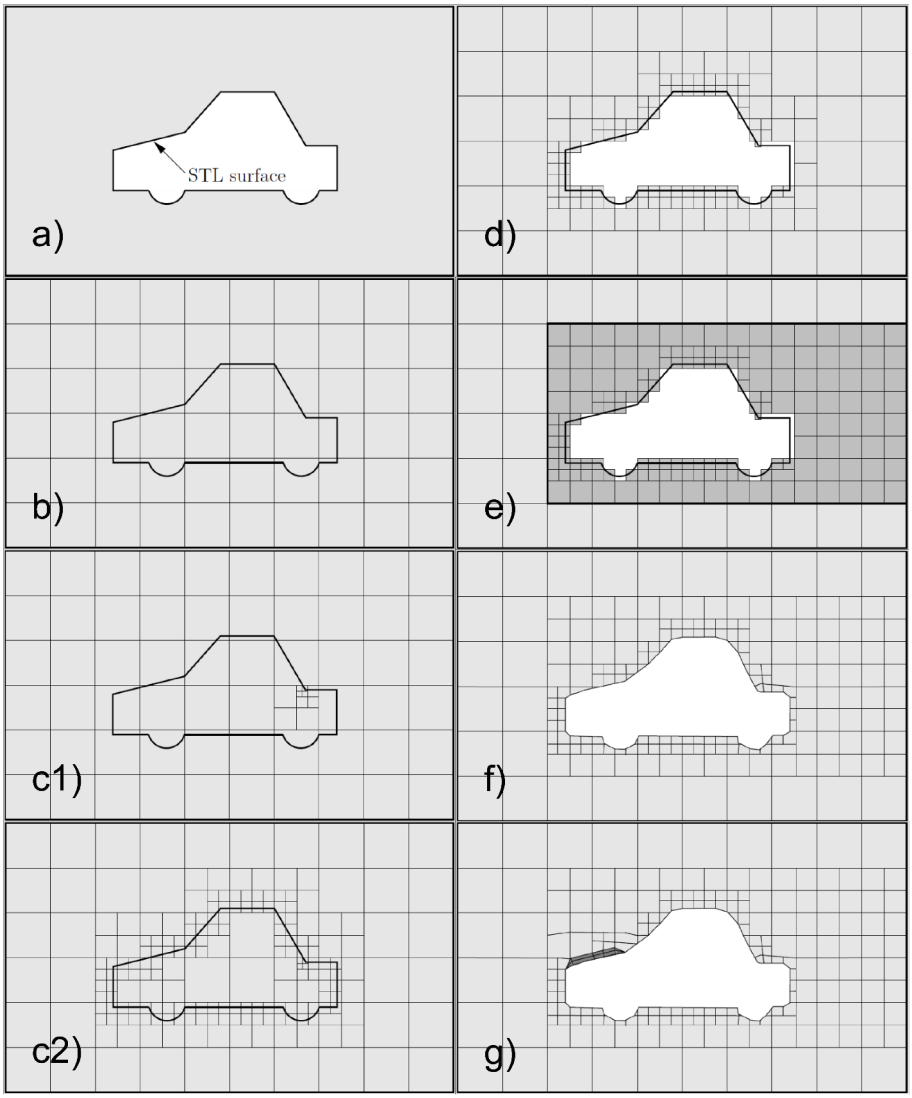


Figure 3.5: *Steps of snappyHexMesh tool - Greenshields [2017]*

4 Analyses in 2D

First simulations were performed in 2D, which are described in this chapter. First, the preliminaries are given in a short introduction. Then, the necessary information regarding the preprocessing is described. Later on, a sensitivity study in regard to several different parameters is given. Finally, the results of 2D analyses for water level, μ_w, and pressures are presented.

4.1 Introduction

Two dimensional models are created with one element in the third direction in *OpenFOAM*. The geometry was based on the initial weir shape with and without the diversion dam. These models were created to survey the basic behaviour of the weir with and without diversion dam and other factors necessary for later 3D computations. The meshes were created with two different methods. First, using the snappyHexMesh algorithm that is included in the basic *OpenFOAM* package. Second, using the external program *Salome*, which is able to create convertible meshes.

The examined points of interest for 2D analyses in addition to the main results were:

- Difference of meshes created with snappyHexMesh and *Salome*
- Mesh size and sensitivity
- Behaviour of the water level with and without the dam
- Influence of friction to the model
- Behaviour of boundary conditions and schemes for later on 3D computations

The geometry was based on a cross section in the middle of the weir. The geometry with its coordinate system is given in Figure 4.1. The zero point of the x-coordinate is situated at the weir's rotation axis for all graphs, the weir front at $x = -5$. Furthermore, the rotation axis and the main measuring axis for comparison with the physical model test (described in Figure 2.10) are shown. The location of the diversion dam varies depending on the cross sections position, as the dam is not parallel to the weir itself. The separated surfaces and their main boundary condition (BC) are given in Figure 4.2. The wall-type area "weir" started at the diversion dams front. If the dam was not included in the model, wall "weir" started at the weir's front and the bottom was stretched.

The 2D simulations were performed in two steps regarding the convection term. First computations were done with a first order `Gauss upwind` scheme, creating a developed initial state. Later on the CD-scheme `Gauss linear` was used for a higher accuracy. All other schemes remain constant within this thesis. The central differencing scheme is unbound and therefore very sensitive to the initial state after the first order computation, especially to the state developed at the inlet. Therefore it was possible to use this scheme for 2D, but not for later 3D computations. The input file for the schemes can be found in the Appendix 8.1, where all chosen settings are shown.

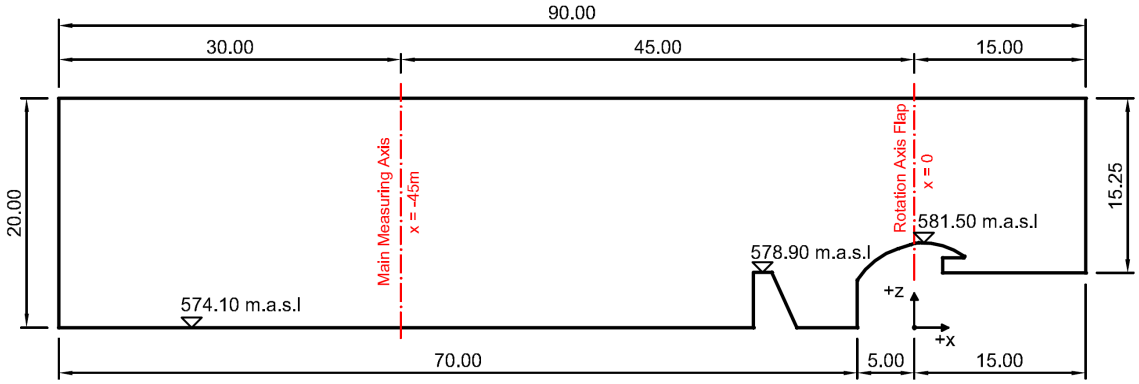


Figure 4.1: *2D Geometry - Outline*

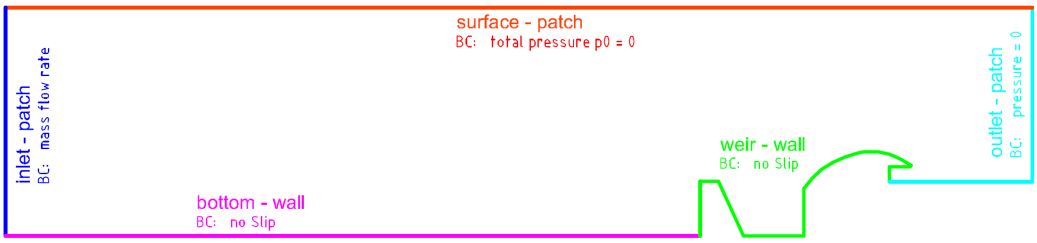


Figure 4.2: *2D Geometry - Patches*

4.2 Preprocessing

With both tools, *OpenFOAM's* snappyHexMesh and *Salome*, the goal was to create mostly structured hexahedral meshes. The experiences made for each tool are described in the following sections.

4.2.1 Meshing process with snappyHexMesh

First experiments of the meshing were done with *Helyx-OS*, which is a user interface for *OpenFOAM* using the snappyHexMesh algorithm. The basics of this algorithm are described in Section 3.5. The full input file with all settings for meshing is included in the Appendix 8.1.

Compared to other meshing tools, snappyHexMesh has no possibility to prepare geometries. Furthermore, it is only possible to create 3D meshes, therefore even for a 2D case, elements are created in the third direction. To create 2D meshes it is possible to extrude the sides to solve this problem and to have meshes with only one element in depth.

Problems that occured were:

- Creation of boundary layers (collapsed around edges)
- Wrongly snapped edges
- Generation of elements not solving the geometry fully, which may cause troubles after extrusion

Some of these problems are shown in the Figure 4.3. Despite all the quality settings possible for mesh, refinement, and layers, those problems are only solvable with a right base mesh size and refinement. To snap edges correctly, it needs a right ratio between the surface length and the element size next to this surface.

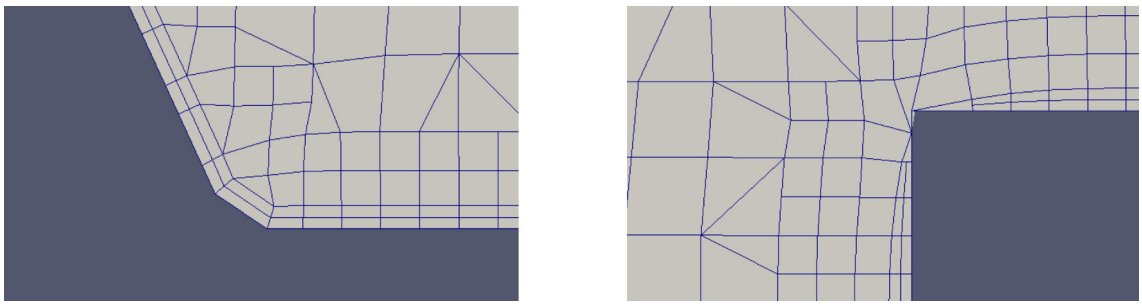


Figure 4.3: *Meshing irregularitys with snappHexMesh*

4.2.2 Meshing process with Salome

Salome is a separate non-commercial program that can be used for mesh generation. It uses different meshing algorithms and allows a direct creation of geometries.

The creation of 2D meshes with *Salome* is rather simple compared to the snappyHexMesh tool. First it is necessary to import the geometry and name the faces for further use. Afterwards a mesh can be created using a main algorithm and sub-meshes in specific order to build the mesh as required.

The meshing process used in this thesis for a 2D hexahedral mesh follows these basic steps:

- Generation of a 3D mesh (Extrusion 3D algorithm)
- Creation of a 2D mesh of one side (NETGEN 1D-2D algorithm)
- Projection of the meshed face to the other one (Projection 1D-2D algorithm with quadrangular preference)
- Filling up the space in between those faces with a single element (Wire Discretisation algorithm)

The mesh size settings are done at the first face to be meshed. In this sub-mesh also the refinement and its general fineness can be chosen. The fineness settings affect growth rate and the number of segments per edge / radius. Further information about the meshing algorithms can be found online in the documentation of Salome [2017].

4.2.3 Comparison of Meshes

In Figure 4.4 the difference of these two ways of meshing are shown at the weir and diversion dam. The top graphic shows a mesh created with *Salome*, the bottom one is created with *OpenFOAM's* snappyHexMesh tool.

Salome creates good computable hexahedral 2D meshes, although the elements are less uniform than in meshes created with snappyHexMesh. A comparison of the meshes with their quality factors created with *OpenFOAM's* `checkMesh` function is given in Table 4.1.

Table 4.1: *Comparison meshing with* `checkMesh`

Factor	***Salome* mesh**	**SnappyHex mesh**
Number of elements	12087	9981
Mesh non-orthogonality max. / average	47.09 / 9.69	33.40 / 5.03
Maximum skewness	2.50	3.01

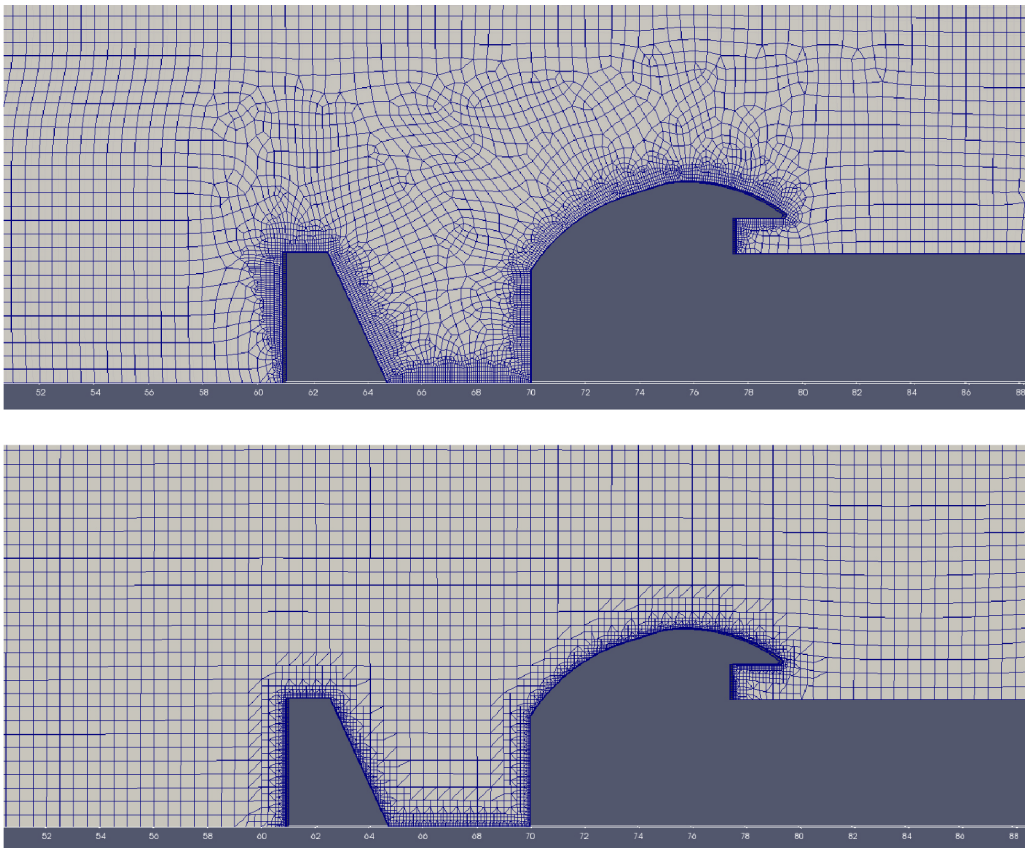


Figure 4.4: *Comparison 2D meshes - Salome / SnappyHex*

Salome has fineness settings that have control over the growth rate of elements along refined areas. The pictured mesh was created with "fine" settings, but even with "coarse" settings the number of elements (10853) was slightly above the mesh created with snappyHexMesh. A comparison of both tools regarding their advantages and disadvantages for 2D meshing is given in Table 4.2.

4.2.4 Boundary Conditions

The used boundary conditions for 2D computations are summarized in Table 4.2.4. The full files are added in the Appendix 8.1.

The inflow was regulated by use of `variableHeightFlowRateInletVelocity`. This boundary condition is especially of interest for multiphase analysis, as it allows to define a certain inflow with a variable phase fraction at the inlet. Discharges were scaled according to the weir's and the model's width. Pressures at the outlet were fixed to 0. This ensures a free outflow of the system. The fixed values for k and ϵ were calculated according to Section 3.3 with a turbulent intensity of 4% and estimated inlet water heights from the physical model test. They are evaluated and checked again in Section 4.5. Both walls were modelled with a no slip condition by use of wall functions. The separation into two parts gave the possibility to add different roughness's later on. The side surfaces were set to empty, so *OpenFOAM* is not solving into this direction.

Table 4.2: *Comparison of tools for 2D meshing*

Meshing tool	
Salome	**SnappyHex**
Advantages: • Graphical user interface to modify geometry • 2D mesh creation possible • Automatic mesh creation following a few inputs • Less dependent on the geometry	**Advantages:** • Included in *OpenFOAM* • Creation of high quality meshes with low non-orthogonality • Elements are aligned to the surface • High control over refinement at defined surfaces • Parallel processing possible • Possibility to use GUIs like *HelyxOS* to ease construction
Disadvantages: • Less control over refinement • Lower non orthogonality • Refinement has a higher influence on the connected domain	**Disadvantages:** • No opportunity to create and modify geometry • No possibility to create 2D meshes straight • Error-prone (e.g. Layer creation) • High dependency on geometry and refinement

Table 4.3: *2D boundary conditions*

Region	Variable	Type
inlet	pressure p	`zeroGradient`
	velocity U	`variableHeightFlowRateInletVelocity - flowRate equals equivalent discharge`
	phase fraction α	`variableHeightFlowRate - lower bound 0.3 / upper bound 0.95`
	turbulent dissipation ϵ	`fixedValue - uniform` ϵ `initial`
	turbulent kin. energy k	`fixedValue - uniform` k `initial`
	turbulent viscosity ν_t	`calculated`
outlet	pressure p	`fixedValue - uniform 0`
	velocity U	`inletOutlet - inletValue uniform (0 0 0)`
	phase fraction α	`zeroGradient`
	turbulent dissipation ϵ	`inletOutlet - uniform` ϵ `initial`
	turbulent kin. energy k	`inletOutlet - uniform` k `initial`
	turbulent viscosity ν_t	`calculated`
surface	pressure p	`totalPressure - p0 uniform 0`
	velocity U	`pressureInletOutletVelocity`
	phase fraction α	`inletOutlet - inletValue uniform 0`
	turbulent dissipation ϵ	`inletOutlet - uniform` ϵ `initial`
	turbulent kin. energy k	`uniform` k `initial`
	turbulent viscosity ν_t	`calculated`
walls:	pressure p	`fixedFluxPressure`
bottom	velocity U	`noSlip`
weir	phase fraction α	`zeroGradient`
	turbulent dissipation ϵ	`epsilonWallFunction`
	turbulent kin. energy k	`kqRWallFunction`
	turbulent viscosity ν_t	`nutkWallFunction`
sides:	pressure p	`empty`
ffminy	velocity U	`empty`
ffmaxy	phase fraction α	`empty`
	turbulent dissipation ϵ	`empty`
	turbulent kin. energy k	`empty`
	turbulent viscosity ν_t	`empty`

4.3 Mesh Sensitivity Study

4.3.1 Preliminaries

To be able to perform 3D simulations, a mesh sensitivity study was carried out using 2D models. This step is necessary as the results of the water levels using two phases, as described in Section 3.1.2, are mesh dependent. The goal of this study was to find a maximum mesh size and refinement to solve the case sufficiently without a huge error on the μ_w value. This variable was only calculated by use of the Poleni formula in Equation 2.1.

These cases were computed for SHQ, with and without the diversion dam, using snappyHexMesh. Five different base mesh sizes were used with a refinement up to level 3 and two added layers around the weir and dam. The main meshes are summed up in the Tables 4.4 and 4.5.

Table 4.4: *2D meshes with diversion dam*

Base size [cm]	Ref. level 3 [cm]	Layer size [cm]	Nr. of Elements
50	6.250	1.8750	9981
35	4.375	1.3125	18423
25	3.125	0.9375	33514
20	2.500	0.7500	50135
15	1.875	0.5625	85564
10	1.250	0.3750	184264

Table 4.5: *2D meshes without diversion dam*

Base size [cm]	Ref. level 3 [cm]	Layer size [cm]	Nr. of Elements
50	6.250	1.8750	8486
35	6.250	1.3125	16224
25	3.125	0.9375	30432
20	2.500	0.3750	46492

4.3.2 Results

There are two main points of interest regarding the results of the mesh sensitivity study. First, the accuracy of computations that was evaluated by comparison of the μ_w coefficients and second, the computation time. These results are now described in more detail.

4.3.2.1 Accuracy of Computations

For water levels at the measuring point for μ_w comparison, values at α = 0.5 were used. These were extracted using a contour plot in *Paraview*, which gave the interface between water and air within the whole domain. A different possibility would have been the use of pressure probes as described in Section 3.1.2. The results for the μ_w value in respect to the base mesh size are given in Figure 4.5. It shows the analysis with remaining diversion dam with second order convection term schemes.

As can be seen, there is a convergence of the μ_w coefficient the finer the mesh gets. This was approximated with a second order polynomial function. The actual result should be around 0.711 for this case. Nevertheless all those results were quite close to this value. Another comparison can be done with the deviation in percent. This is shown in Figure 4.6, the results of the finest mesh are used as base. The differences regarding μ_w were beneath 1%, so even with a coarse mesh it is possible to achieve appropriate results. There were only minor differences between the results of first and second order computations. Accepting an error of approximately 1-3% would also allows to use a first order convection term scheme for computations with these mesh sizes.

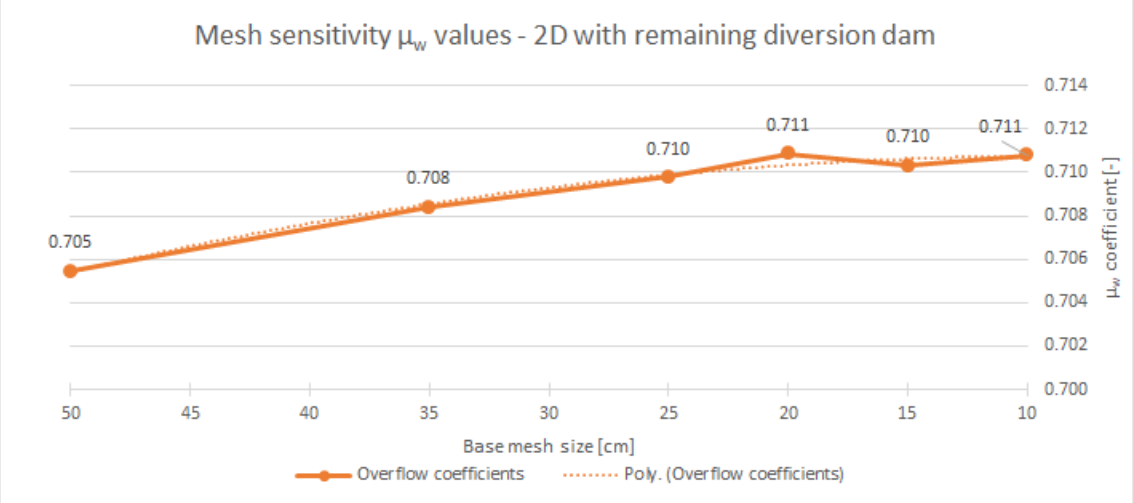


Figure 4.5: *Mesh sensitivity study in 2D - μ comparison*

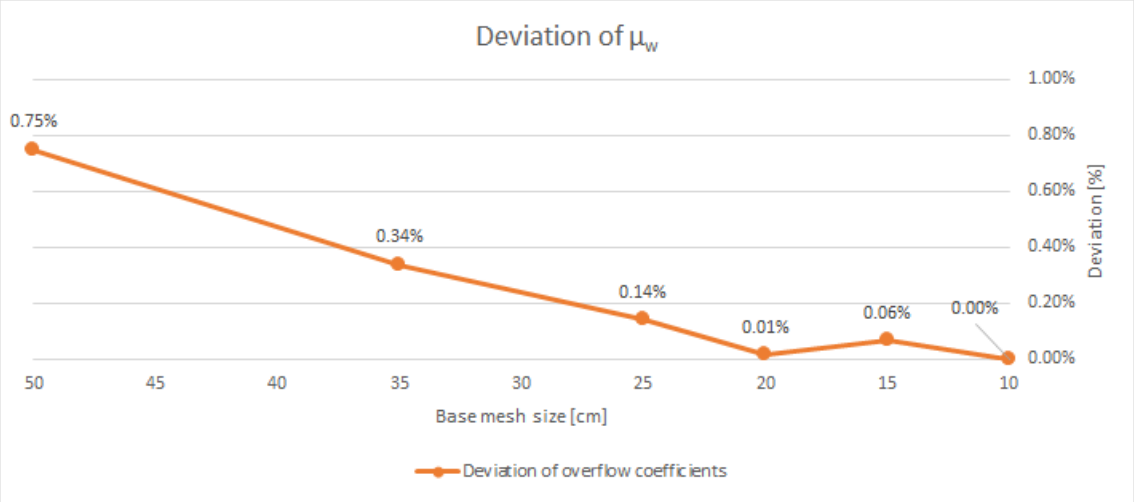


Figure 4.6: *Mesh sensitivity study in 2D - Deviation*

4.3.2.2 Computation Time

Another important point regarding the mesh size is the necessary computation time. As these cases were in 2D, the computation time will increase massively for the 3D simulations. There are two variables that need to be observed in relation to the time, these are the time step Δt and the necessary time for each iteration.

The iteration time highly depends on the used hardware. Crucial parts for analysis with *OpenFOAM* are CPU, RAM and hard-drive. The used computational setup is given in Table 4.6.

Table 4.6: *Hardware Specification*

Part	Product Name	Description
CPU	Intel Xeon X5650	6 physical 2.67GHz cores with dual threading
RAM	-	6 times 4GB DDR3 (1333 MHz)
Hard-drive	SanDisk SDSSDH12	120GB Solid State Drive

The key variables, Δt and iteration time, can be found in *OpenFOAM's* log files created during runtime. The numbers were taken after the computation reached a stable state. Whereas the necessary iteration time increases exponentially with a finer mesh, the time step Δt decreases linearly. This linear decrease of the step happens due to the adjustable time stepping of *OpenFOAM*, which reduces the step according to the Courant Number given in Equation 4.1 for a 1D case. For each direction another term is necessary. The Courant Number has to be below 1.

$$C = \frac{u\Delta t}{\Delta x} \leq C_{max} \tag{4.1}$$

The velocity u can be seen as constant at a stable state. Therefore, the time step Δt varies linear with the distance between computation points Δx, which was the base mesh size within the domain for these cases. The extracted data is shown in Table 4.7.

Table 4.7: *Mesh sensitivity study in 2D - Computation time*

Base size [cm]	**Δt [s]**	**Iteration time [s]**	**$\frac{\text{Iterationtime}}{\Delta \text{t}}$**	**Normalized**
50	0.0013	0.010	7.77	0.011
35	0.0009	0.015	16.40	0.024
25	0.0007	0.030	44.16	0.064
20	0.0006	0.040	66.96	0.098
15	0.0005	0.080	171.03	0.249
10	0.0003	0.235	686.05	1.000

The value of $\frac{Iterationtime}{\Delta t}$ gives an approximation of the time that is necessary for computing one second of real time with the used hardware. Therefore it contains both mesh dependent effects and can be used to validate the overall computation time. This value was normalized and plotted in Figure 4.7.

The computation of the finest used mesh took almost 100 times longer than for the coarsest one. The number of elements had increased approximately by a factor of 18.5. As the iteration time increases exponentially, the difference gets enormous.

4.3.3 Summary

As the results of Section 4.3.2.1 had shown convergence, meshes even finer than 10 centimetres were not analysed. The water level may vary within a range of a few centimeters, but the μ_w coefficient is less sensitive to this change than a direct comparison. Including the findings of the computation time comparison, which had shown a massive increase of necessary time for fine meshes, a base mesh size of 35 centimetres was chosen.

This mesh size had a deviation of less than 0.35% regarding the μ_w value and a computation time which was approximately 40 times faster than for the finest mesh. Even though the error will increase in a 3D model, the massive difference in the necessary time justifies this choice, which will also increase at 3D runs. Further on all models will have this base mesh size.

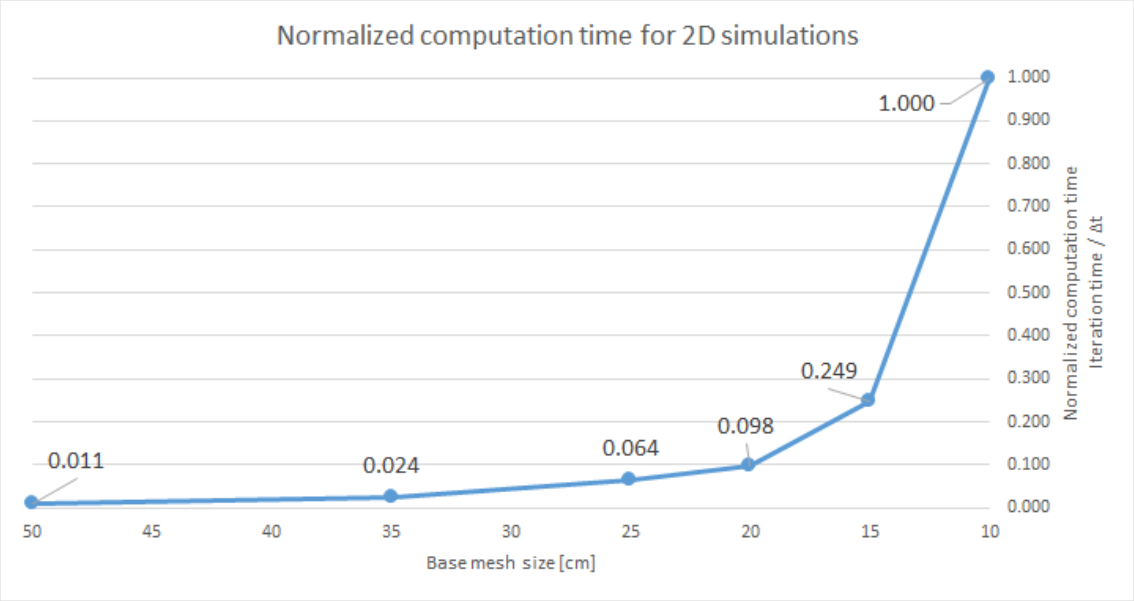


Figure 4.7: *Mesh sensitivity study in 2D - Normalized computation time*

$$Chosen\ base\ (maximum)\ mesh\ size = 0.35m \tag{4.2}$$

4.4 Influence of Roughness

As the mesh size was fixed it was possible to properly examine the behaviour of wall functions. First, y^+ described in Section 3.4 was evaluated at the first node next to the reservoirs bottom. This was done with and without layers. Relative layer sizes of 30% and 10% to their surrounding mesh were used. Below a minimum layer size of 10% the layer creation of snappyHexMesh got unstable.

The observed cases are given in Table 4.8. The first value describes the bottom layer, whereas the second one represents the layer around the weir with its refinement. The distance from the wall to the first node is also of interest for the maximum possible roughness to add. The minimum layer thickness should be the half sand grain roughness k_s that is added, otherwise this would create errors. The results for a 2D SHQ model with the diversion dam from inlet to dam are plotted in Figure 4.8.

Even without the use of layers, the y^+ values were within the range of 11.63 and 300. Nevertheless this result almost reached the upper border, so a local increase of the velocity may lead to a violation of the limit. Adding layers led to lower and more constant numbers. Layers with 30% of the neighbouring elements size were chosen, as these meshes had a rather smooth transition from the layers to the next cell. Furthermore, the addable k_s values were sufficient.

Roughness was added to this mesh by use of the wall function `nutkRoughWallFunction` instead of `nutkWallFunction` used for smooth walls . The input parameter for roughness

Table 4.8: *Layer sizes for roughness study*

Relative layer thickness bottom / weir	**Absolute layer thickness [cm] bottom / weir**	**Maximum k_s [mm] bottom / weir**
No Layer / 0.3	35 / 1.3125	175 / 6.5625
0.3 / 0.3	10.5 / 1.3125	52.5 / 6.5625
0.1 / 0.1	3.5 / 0.4375	17.5 / 2.1875

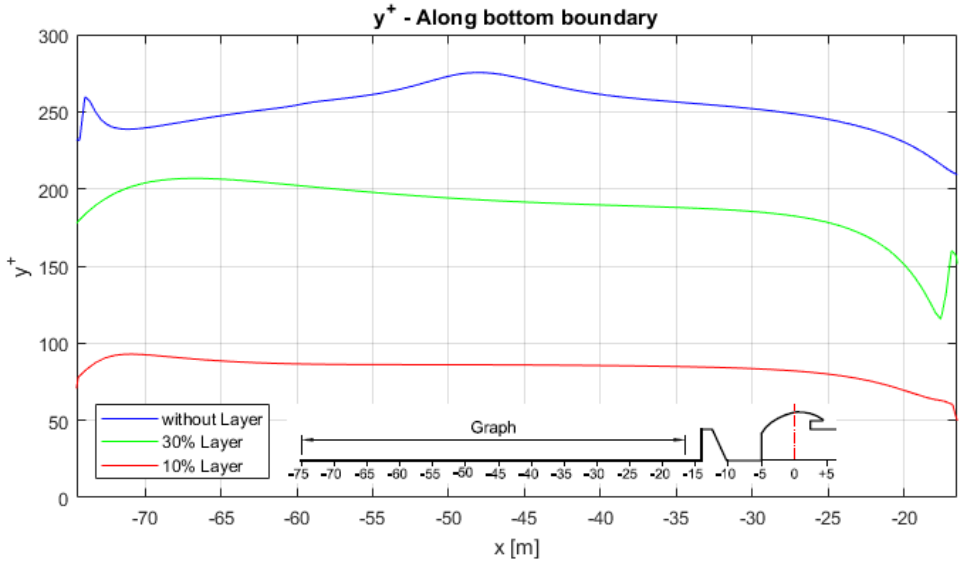


Figure 4.8: *Roughness study - yPlus along bottom boundary for SHQ*

in *OpenFOAM* is the equivalent sand roughness k_s which can be found for modelling and natural material in literature such as Kirschmer [1974] and Rössert [1994]. The observations were done for a model without the dam, as the bottom is longer and the roughness in this area has a bigger influence. Two additional cases were compared in addition to the smooth one by use of a SHQ discharge. The input data is given in Table 4.9.

Table 4.9: *Roughness - k_s values*

Test case	k_s bottom [mm]	k_s weir [mm]
smooth	-	-
Altered model k_s	2	0.0015
Natural k_s	40	4

As described in Section 2.2.2, the model test was created mainly with PVC, which can be considered smooth. The first rough case would represent an alteration of the model, for example by adding fine grained sand to the upstream channels bottom. The second case should represent a more natural state with gravel at the bottom and worn of concrete at the weir. These k_s values were within the possible range of the selected layers of 30% given in Table 4.8. The evaluation of results was done by comparison of the water levels for SHQ from the measuring point to the front wall of the weir shown in Figure 4.9.

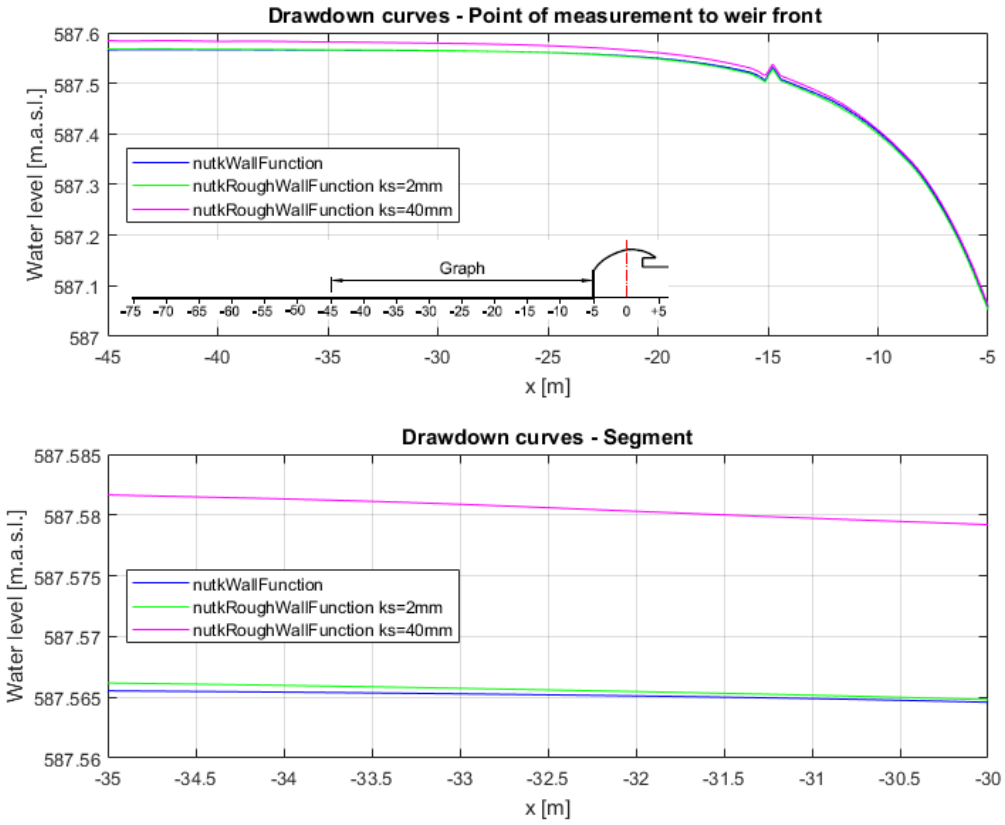


Figure 4.9: *Roughness - Change of drawdown curves for SHQ with addition of roughness*

In the upper graph of Figure 4.9 the difference in water height regarding roughness is hardly to see. It is necessary to use a smaller segment to make the increase of the water levels due to higher roughness visible. There was almost no change from the smooth case to the altered model roughness, but using more natural values resulted in a minor change. This increase of approximately two centimetres resulted in a decrease of μ_w of about 0.003. As this change was rather small, the effect of roughness regarding the water level can be neglected. The peak at around 60 meters was caused by a row of skew elements produced by the layer creation at the bottom and should be ignored.

Another point that was surveyed during this roughness study were the shear stresses along the walls. These are plotted from the main measuring point to the weir's front in Figure 4.10 for a SHQ discharge.

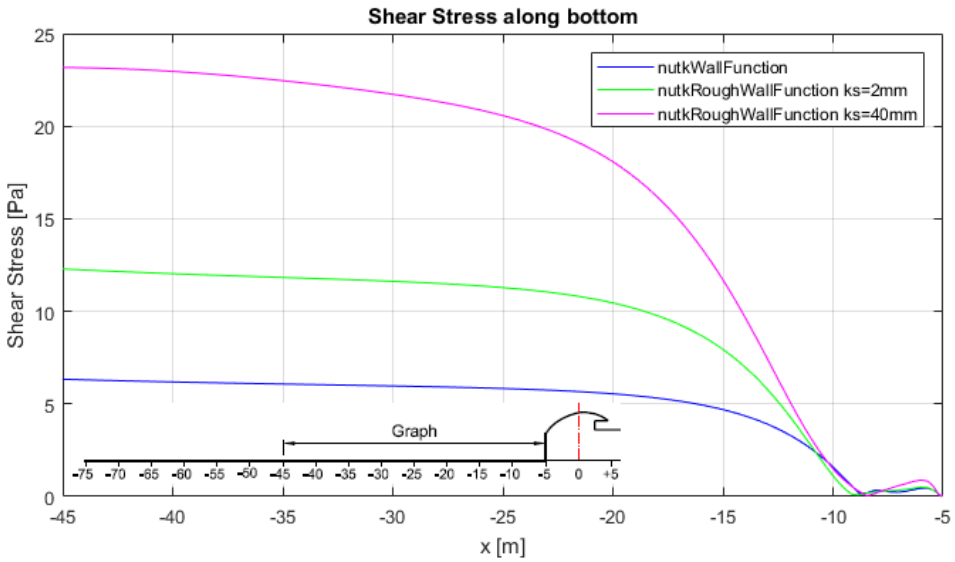


Figure 4.10: *Roughness - Shear stress along bottom boundary for SHQ*

This graph shows a significant difference of the shear stress with added roughness. If effects influenced by the shear stress would be observed, it would be necessary to add roughness. As the main points of interest in thesis were the water levels and the overflow coefficients μ_w, roughness was ignored for 2D analyses.

4.5 Turbulence Parameters

With data acquired from previous computations, the input parameters for k and ϵ were validated using the experimental formulas mentioned in Section 3.3. A comparison of the calculated values with both methods is given in Figure 4.11. The shear velocity is calculated as in Equation 3.30 with the bottom shear stress τ_w. A value of six Pascal within the domain was selected from Figure 4.10. Only values of the SHQ computations are shown, but due to a decrease of velocity and shear stress for BHQ, the comparison of the turbulence parameters shows the same relation.

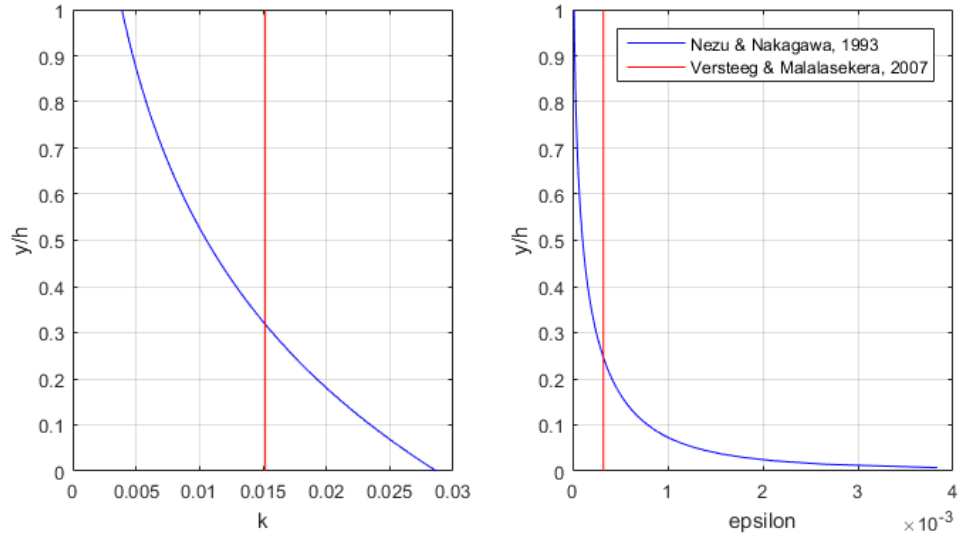


Figure 4.11: *Turbulence validation - Comparison simple formula and experimental data*

The experimental data shows the distribution of k and ϵ that would occur in an open-channel flow in a fully developed state. All models computed within this thesis had obstacles, so this natural state will hardly develop. As the constant values showed a good approximation with the experimental data, the simple formulas from Versteeg and Malalasekera [2007] and constant input parameters were used. Nevertheless, it has to be noted that added roughness may change the curves significantly as the friction velocity $u_\star$ increases. The final input values for k and ϵ calculated with the simple formulas and an approximated inlet heights are given in Table 4.10.

Table 4.10: *Turbulence parameters for 2D analysis*

Case	$\mathbf{H_{Inlet}}$	**k**	$\boldsymbol{\epsilon}$
BHQ - With diversion dam	12.0	0.006567042	0.000104091
BHQ - Without diversion dam	11.8	0.00679154	0.000111329
SHQ - With diversion dam	13.8	0.014525551	0.000297755
SHQ - Without diversion dam	13.5	0.015178305	0.000325118

4.6 Results

At the final stage of the 2D analysis both models were computed with the final mesh and input parameters for BHQ and SHQ. The results of this computations were compared to results of the physical model test regarding the water level and the μ_w coefficient. Furthermore pressures along the weir of all computed cases were analysed.

4.6.1 Water Level / Overflow Coefficient

First, the water levels at and the resulting μ_w of the two CFD models are compared at the measuring point in Table 4.11. A comparison with the physical model test was done using the available data from Table 2.3. This comparison is shown in Figure 4.12. As for the physical model tests report (IWB [2014]), this comparison was done with the Poleni formula. Considering mean upstream velocities of approximately 1.7 m/s for BHQ and 2.5 m/s for SHQ, would suggest the usage of the Weisbach formula from Equation 2.2. This would lead to a decrease of μ_w of approximately 0.02 for BHQ and 0.04 for SHQ.

Table 4.11: μ_w *- Comparison of 2D CFD*

2D model	$\mathbf{H_{BHQ}}$ **[m.a.s.l.]**	$\boldsymbol{\mu}_{\mathbf{w,BHQ}}$	$\mathbf{H_{SHQ}}$ **[m.a.s.l.]**	$\boldsymbol{\mu}_{\mathbf{w,SHQ}}$
With diversion dam	585.92	0.72	587.91	0.71
Without diversion dam	585.92	0.72	587.56	0.77

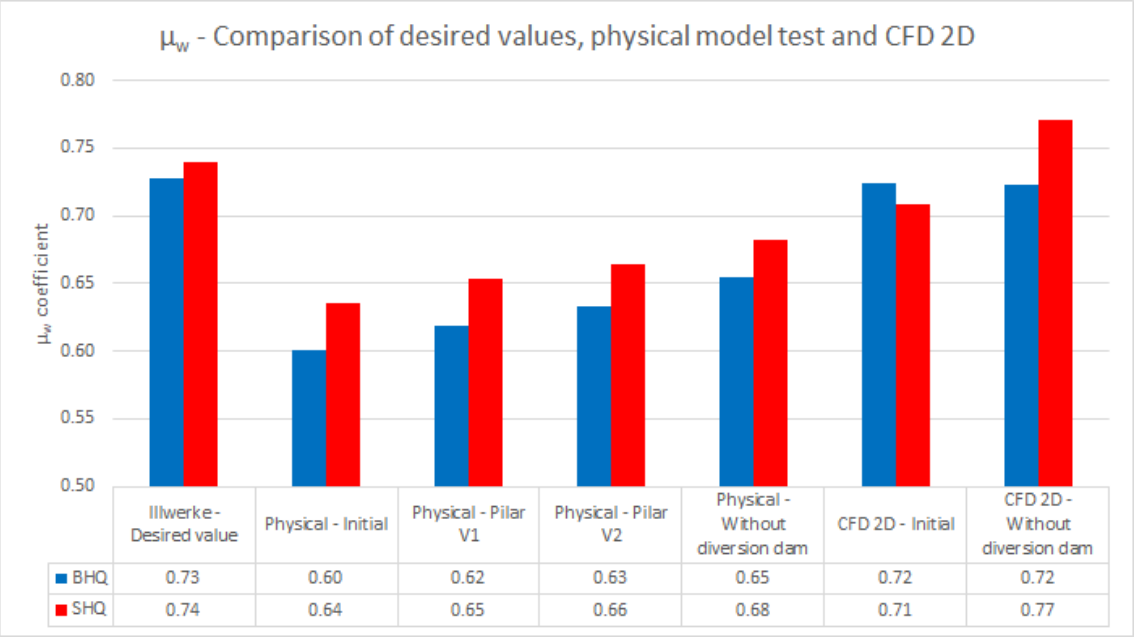


Figure 4.12: μ_w *- Comparison of desired values, physical model test and CFD 2D*

If neglecting the operator's desired values, the comparison clearly shows that the μ_w coefficients of the physical model test are below the results of 2D CFD analyses. One of the reasons for this is the missing contraction due to the three dimensional flow in the physical test. A further interesting point was, that in the physical model test the values of $\mu_{w,BHQ}$ were always below $\mu_{w,SHQ}$. This is not showing at the 2D analysis containing the diversion dam.

In Figure 4.13, the drawdown curves are compared from the measuring point to the weir's axis. For this scenario full data from the physical model test was only available for a model with an already removed diversion dam. Thus, there is only one result for BHQ and SHQ each.

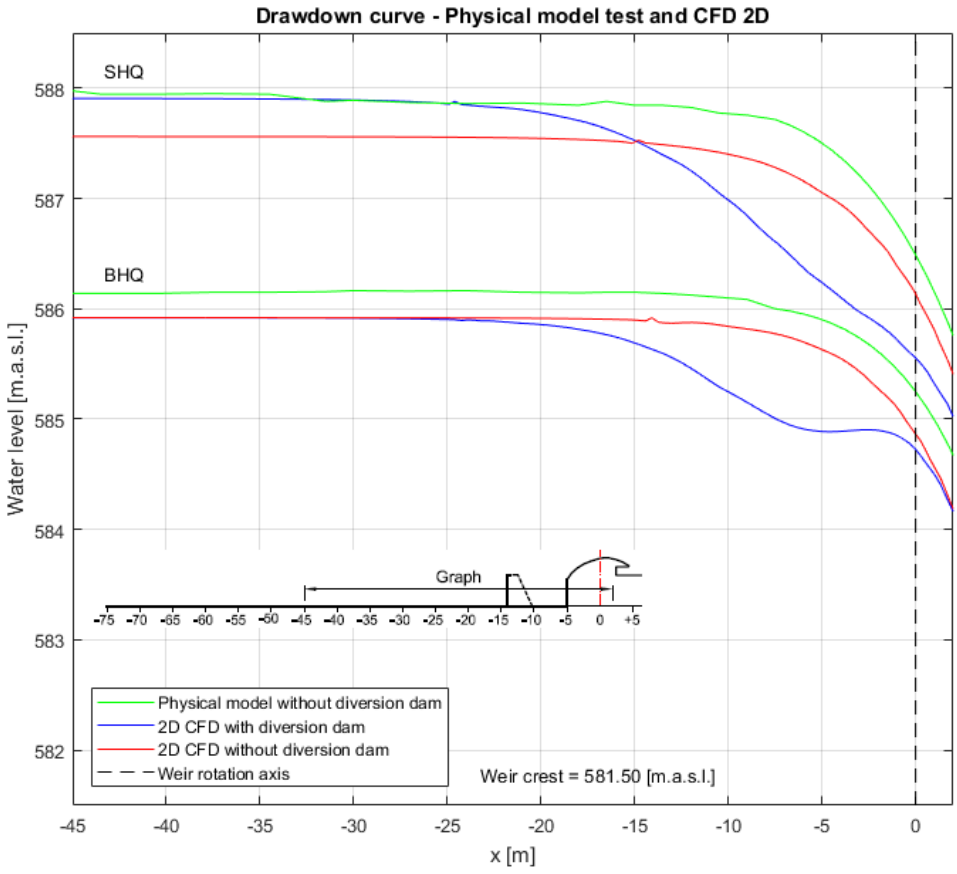


Figure 4.13: *Drawdown curves - Physical model test and CFD 2D*

At the weir the flow changes from sub- to supercritical. The location of this change can be found with the critical water height h_{crit} and the acquired water levels from the simulations. As shown in Equations 4.3 and 4.4, the critical water height can be calculated with the energy height of the approach flow h_{min} and the relation of h_{crit} to h_{min} for a rectangular channel. The input parameters and the calculated h_{min} can be found in Table 4.12, the location of those water levels in Table 4.13. The x-coordinate is again dependent on the weir's rotation axis, the water level is given with the geodetic height.

$$h_{crit} = \frac{2}{3} \cdot h_{min} \tag{4.3}$$

$$h_{min} = h_0 + \frac{v_0^2}{2g} \tag{4.4}$$

Table 4.12: *Critical water heights*

Case	$\mathbf{h_0}$ **[m]**	$\mathbf{v_0}$ **[m/s]**	$\mathbf{h_{crit}}$ **[m]**
BHQ - With diversion dam	4.42	1.68	3.04
BHQ - Without diversion dam	4.42	1.68	3.04
SHQ - With diversion dam	6.41	2.46	4.48
SHQ - Without diversion dam	6.06	2.52	4.26

Table 4.13: *Location of critical water heights*

Case	**Location - x [m]**	**Water level [m.a.s.l.]**
BHQ - With diversion dam	0.81	584.54
BHQ - Without diversion dam	1.09	584.53
SHQ - With diversion dam	-0.24	585.68
SHQ - Without diversion dam	1.15	585.75

4.6.2 Pressure Distribution

Finally, the pressure distribution along the weir's top surface was compared for the four computed cases. A confrontation of the same flow with and without diversion dam is given in Figure 4.14 for BHQ and Figure 4.15 for SHQ. In both figures the upper graph contains the weir's shape, this allows to track the location of those pressures. In addition to the pressures, also the water levels and the location of the critical water heights h_{crit} are included in the upper graphs. Furthermore, the minimum and maximum pressures are given separately for each case in Tables 4.14 and 4.15. The maximum pressures were all situated right at the weir's beginning within the concrete area, whereas the minimum pressures were located at the end of the lowered flap.

Table 4.14: *Pressure distribution along the weir - Maximum values*

Case	$\mathbf{p_{max}}$ [kPa]	$\mathbf{p_{max}}$ [mwc]	**Location - x [m]**
BHQ - With diversion dam	63.38	6.47	-4.91
BHQ - Without diversion dam	70.81	7.23	-4.94
SHQ - With diversion dam	74.63	7.62	-4.78
SHQ - Without diversion dam	83.70	8.55	-4.91

Table 4.15: *Pressure distribution along the weir - Minimum values*

Case	$\mathbf{p_{min}}$ [kPa]	$\mathbf{p_{min}}$ [mwc]	**Location - x [m]**
BHQ - With diversion dam	-4.25	-0.43	4.01
BHQ - Without diversion dam	-4.37	-0.45	4.01
SHQ - With diversion dam	-14.56	-1.49	3.34
SHQ - Without diversion dam	-11.06	-1.13	3.74

It is clear to see that a dam removal decreases the tensional pressures occurring at the lowered flap. At BHQ the diversion dam's influence was small whereas it had a higher impact at the SHQ cases. Right after the flaps rotation axis a local change in pressure was showing, as the geometry has a change of the slope there.

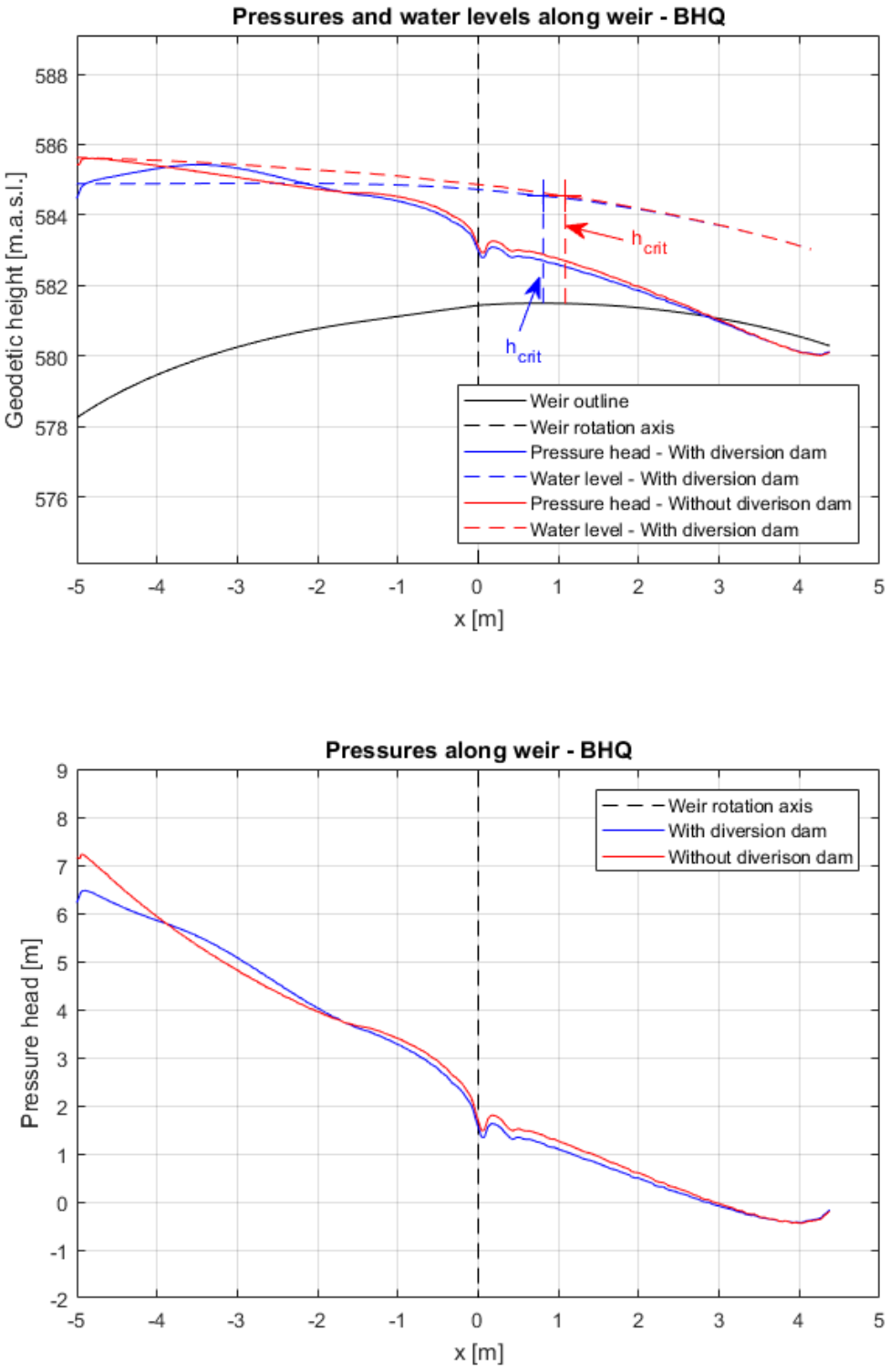


Figure 4.14: *Pressure distribution along the weir - BHQ*

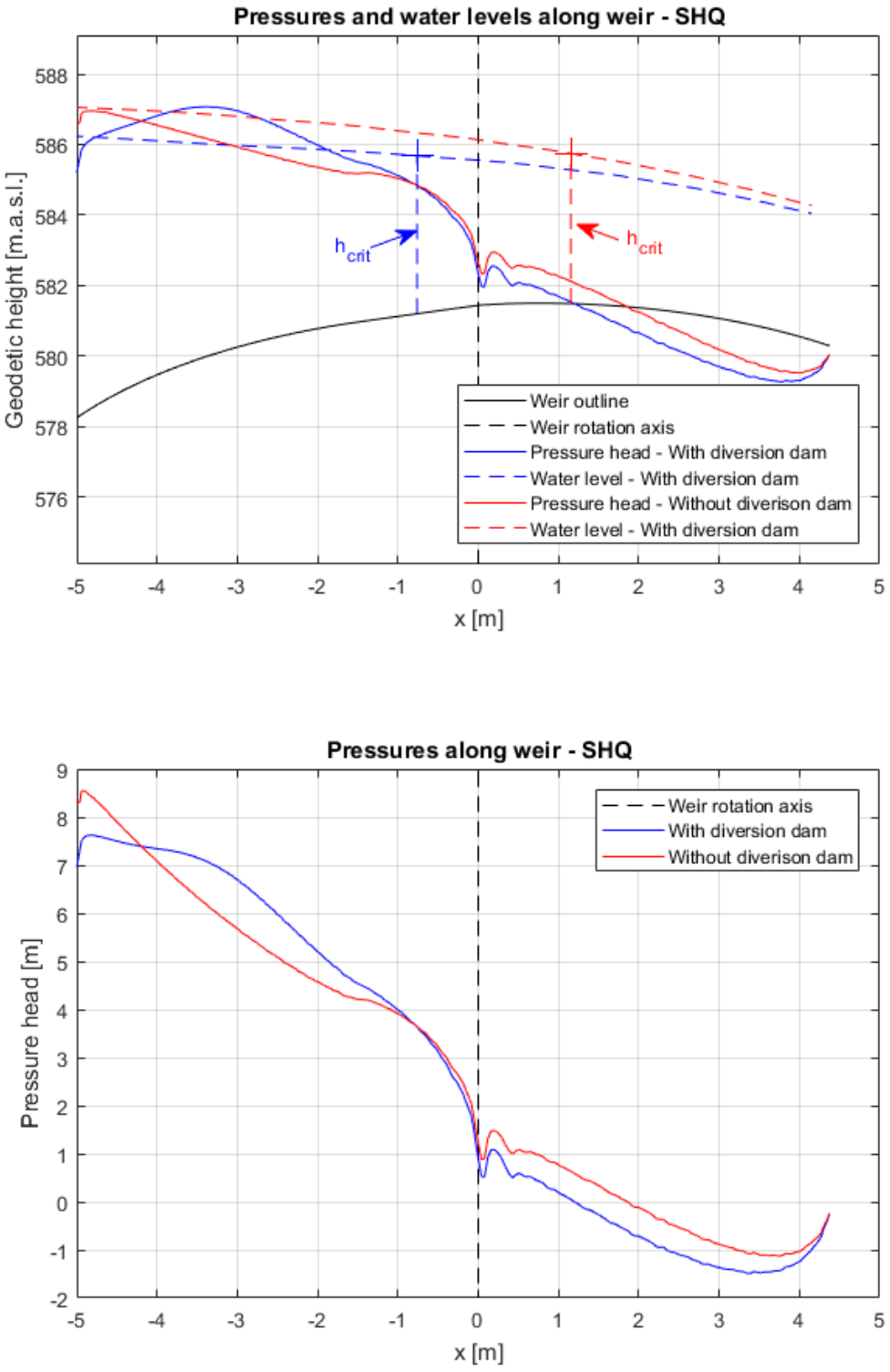


Figure 4.15: *Pressure distribution along the weir - SHQ*

4.6.3 Streamline Comparison

To make the behaviour of a dam removal visible, some stream lines of both systems were compared for SHQ. A removal of the diversion dam had a large impact on the flow behaviour around the weir. As can be seen in Figure 4.16, the stream detaches earlier from the bottom, thus leading to an acceleration of the stream before reaching the weir. The dam caused recirculation that can be seen due to round shape of increased velocity between dam and weir front. The increased velocity led to a drop of the water level around the weir, but to a higher water level before the weir. Thinking about a 3D simulation the dams location with respect to the weir's axis is not constant, as it is non parallel. This will lead to a three dimensional flow in this area.

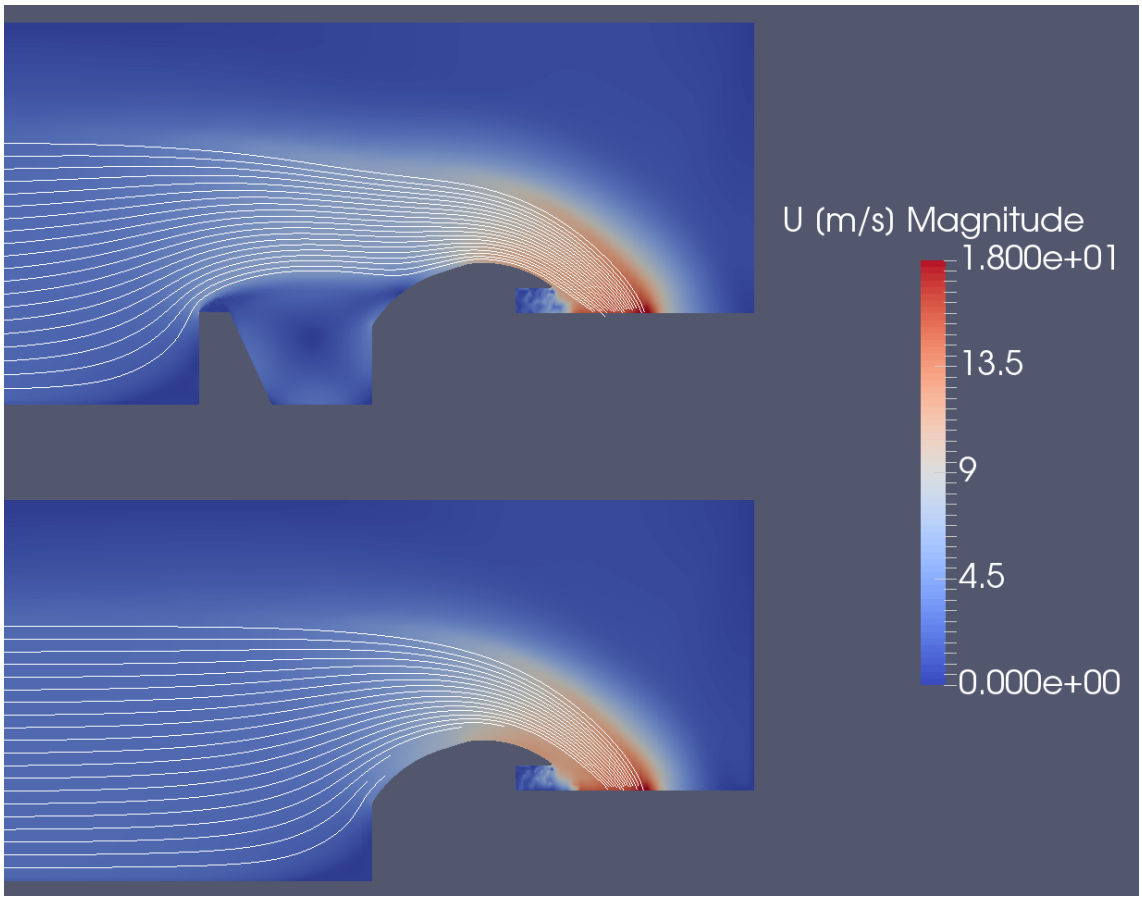


Figure 4.16: *Stream line comparison of 2D models*

4.7 Summary

The 2D studies show, that the preprocessing has a major importance at CFD analyses. There are different tools available to create meshes that are computable with *OpenFOAM*, but it has shown that each one has its advantages and disadvantages. Whereas the external *Salome* tool creates usable 2D meshes without much effort, the usage of *OpenFOAM's* snappyHexMesh is rather difficult for coarse meshes and slightly more complicated geometry but produces high quality meshes.

The mesh sensitivity study showed that very fine meshes are not necessary if the main interest is the determination of μ_w coefficients of weirs. The water level may vary slightly, but the μ_w value is less sensitive to these changes. Even for the coarsest used mesh, the change of accuracy was less than 1% compared to the finest mesh. This deviation may get higher for 3D cases. If a more precise determination of the water level is necessary, a very high increase of the computation time has to be taken into account.

In the 2D computations roughness did not play a huge role regarding the water level. Despite of increasing shear stresses it had no significant influence on the results. Nevertheless, if analyses regarding sediment transportation are necessary, this effect is of great importance. Furthermore, the determination of the turbulence parameters k and ϵ depends on the shear stress and therefore the roughness when experimental formulas are used. Yet, the simple formulas showed a good fit for those 2D cases.

A removal of the diversion dam had no great influence on the BHQ flow, whereas the impact at a SHQ state was even higher than in the physical model. Water levels of all 2D simulations were lower compared to the physical model. This was quite clear as the weir Gstins is a three dimensional weir where contraction plays a role.

The pressure analysis showed that a removal of the diversion dam would have a positive influence on the negative pressures occurring at the flap. In general the obtained pressures for SHQ had higher minimum and maximum values than for BHQ, larger pressures up to the rotation axis and lower ones afterwards.

Based on insights of the 2D simulations some further assumptions for later on 3D modelling were possible. As it has shown during these analyses, the length of channel had to be increased as the system was very sensitive to the initial conditions that were set. Badly defined starting water levels and channel velocities led to an influenced measuring point. Nevertheless, the general boundary conditions seemed to be sufficient for later simulations.

5 Analyses in 3D

As the weir Gstins is a three dimensional case, it was necessary to investigate the behaviour of a 3D simulation to allow a comparison to the physical model. As in the 2D analyses from Chapter 4 some preliminaries are given, followed by a section dealing with the meshing and preprocessing. A short sensitivity study regarding some parameters is validating the findings from 2D analyses. In the final step, all available results for water level, μ_w and pressures are compared.

5.1 Introduction

The geometry was created based on plans of the physical model test that can be found in the Appendix 8.3. It was modelled for the initial weir shape with the different pillar shapes. This is shown in Figure 5.1. The model is shown with the original pillar form on the orographic right side and the elliptic modification on the left side. Again the zero point of the x-coordinate is situated at the weir's rotation axis in this geometry.

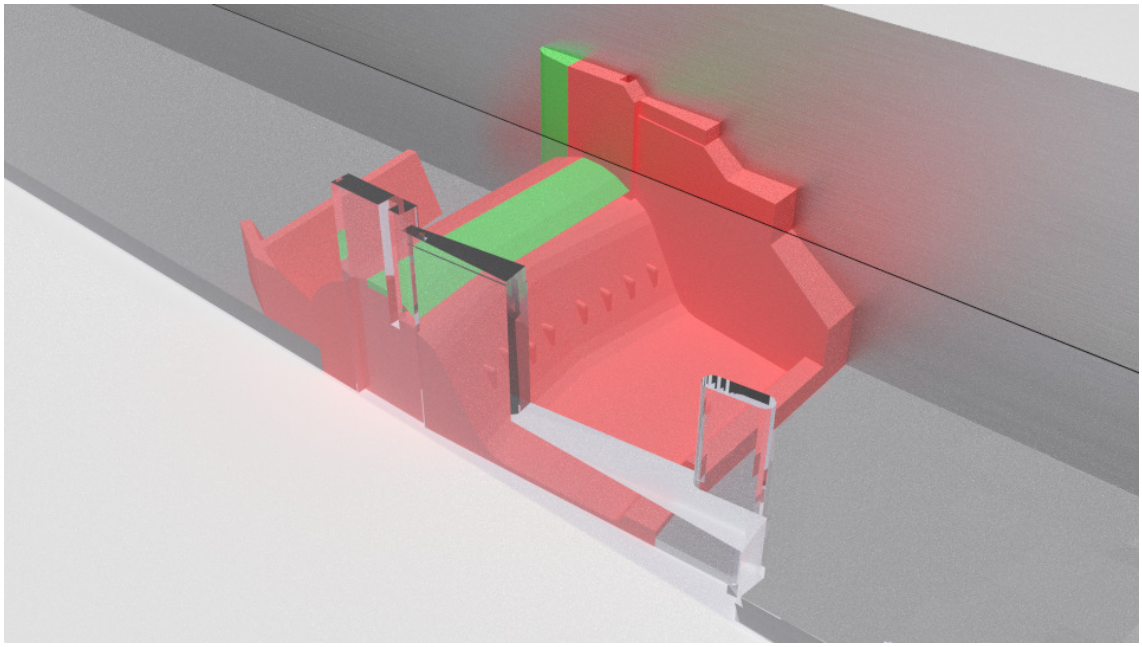


Figure 5.1: *3D - Full geometry*

The examined points of interest for 2D analyses in addition to the main results were:

- Difference of meshes created with snappyHexMesh and *Salome*
- Mesh size and sensitivity
- Comparison of non-symmetrical and symmetrical models
- Influence of roughness to the model
- Behaviour of contraction

5.2 Preprocessing

As the meshing behaviour in 3D is different as in 2D, there was again the choice between two different tools as in Section 4.2.

5.2.1 Meshing process with snappyHexMesh

As the main purpose of the snappyHexMesh utility is the creation of 3D hexahedral meshes, some of the disadvantages mentioned in Table 4.2 disappear. As it was deemed acceptable that some elements do not completely full-fill the geometry at the intersection of side pillar and weir, less refinement was needed. Such elements were not acceptable in 2D, as their extrusion caused problems. Furthermore, the flaps tip was not snapped completely correct, leading to a slight change of geometry in this area. The problematic elements created with a coarse snappyHexMesh are displayed in Figure 5.2. Apart from these elements and changes the mesh showed a good overall structure.

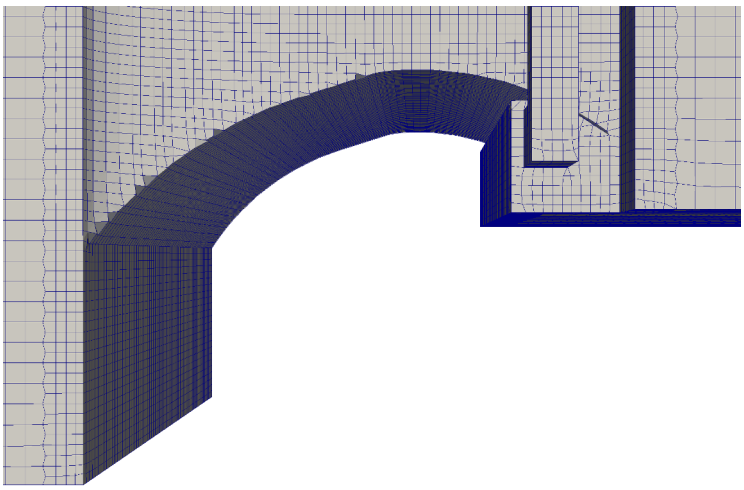


Figure 5.2: *3D meshing - snappyHexMesh detail weir*

5.2.2 Meshing process with Salome

Whereas *Salome* had its advantages due to its easy usage for 2D cases, this is not the case for more complex three dimensional geometry, where hexahedral elements are of interest and simple 2D face mesh extrusion is not possible.

For more complex 3D-hexahedral meshes in *Salome* there are two main possibilities without commercial software. Namely the basic 3D hexahedral meshing algorithms of *Salome* are:

- Body Fitting
- Hexahedron (i,j,k)

The Body-Fitting algorithm is easy to handle if only straight geometry is used, but it makes refinements along surfaces hardly possible. The Hexahedron algorithm allows the use of sub-meshes defining surfaces. Although this method works well for easy geometries, it is rather complicated for more complex situations. The algorithm requires a hexahedral input geometry such as *OpenFOAM's* `blockMesh`, making it necessary to split it up before importing.

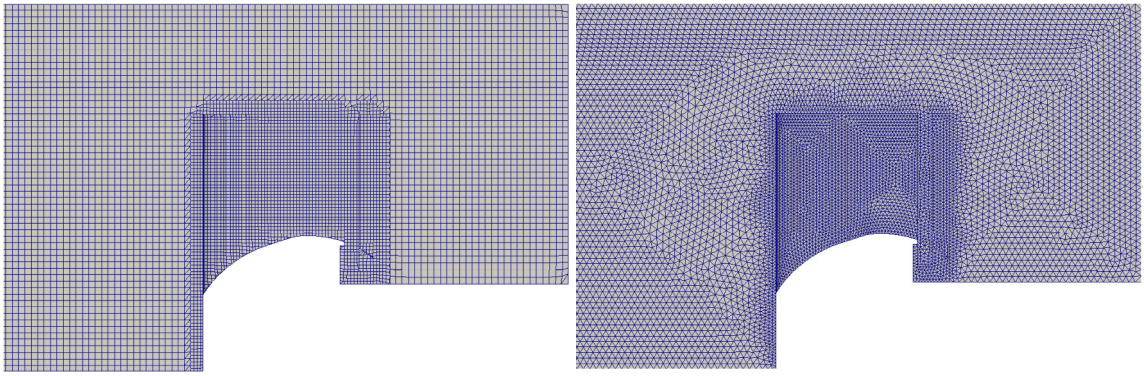
The quite handy NETGEN-3D algorithm mainly works with tetrahedral elements which are not recommended for multiphase analysis as those create an unstructured mesh all over the domain. Nevertheless it was the only algorithm that was able to solve the given geometry without massive effort. Therefore the comparison of mesh creation with snappyHexMesh and *Salome* was done with this algorithm.

5.2.3 Comparison of Meshes

Details of the created cases for comparison are given in Table 5.1. These meshes did not contain the diversion dam and used symmetry in the middle. As tetrahedral elements increase the total number of elements, two meshes with different sizes were created with *Salome* to visualize the influence. A visual comparison of three dimensional meshes created with both tools is given in Figure 5.3.

Table 5.1: *3D meshes*

Meshing tool	**Element length [cm]**	**Refined length [cm]**	**Nr. of Elements**
snappyHexMesh	35	17.5	711916
Salome (NETGEN 3D)	50	17.5	1246357
Salome (NETGEN 3D)	35	17.5	4013643



(a) 3D meshing - snappyHexMesh (b) 3D meshing - *Salome* NETGEN 3D

Figure 5.3: *3D meshing - Comparison of tools*

The comparison of those meshes showed that the tetrahedral *Salome* mesh had a clear disadvantage due to its high number of elements. This would lead to a rise in computation time. Furthermore, the tetrahedral mesh did not create a structured mesh within the domain like snappyHexMesh did. Another noteworthy point is that refinement along the weir had a higher influence on the domain, but *Salome* was able to fully solve the geometry, even at the weir's tip. In terms of computation the tetrahedral mesh led to stability problems in this multiphase analysis.

SnappyHexMesh worked well for the given case besides of the small areas at the weir's side and tip. The hexahedral mesh performed well in computations. Note that a mesh creation for the model with diversion dam would need a higher refinement to solve the geometry properly, leading to an increasing element number.

Summarizing, in this case the snappyHexMesh had a clear advantage to the tetrahedral *Salome* mesh. Therefore it was decided to stick with *OpenFOAM's* own mesh generation tool. Another comparison of those tools can be found in Kortelainen [2009]. In Figure 5.4 the main mesh used for further computations is shown.

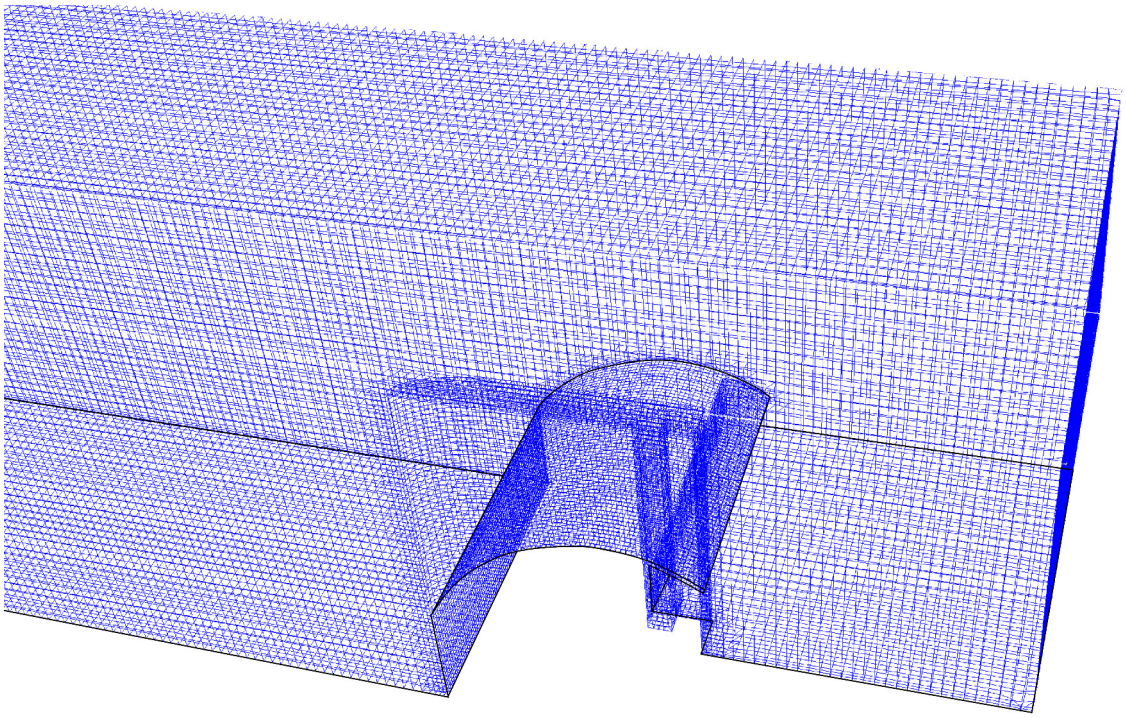


Figure 5.4: *3D - Basic mesh*

5.3 Sensitivity study

As for the 2D cases a sensitivity study for some parameters was performed. The basic mesh for 3D computations was based on the geometry with epileptic pillar extension, had an element size of 0.35 metres, and a level 1 refinement at the weir. The model was considered smooth and used a `symmetryPlane` boundary condition in the middle. Sensitivity variations to this basic model are given in Table 5.2.

Table 5.2: *3D sensitivity study - Variations to basic model*

Modification	**Description**
Symmetry boundary condition	Full model test without `symmetryPlane` boundary condition
Mesh size	Decreased element size to 0.25 metres and increased refinement to level 2 around the weir
Roughness	Added roughness with the higher k_s values from Table 4.9

Due to the high computation times only models without diversion dam were fully computed. These had the advantageous property of symmetry. Only the basic symmetrical

computation was performed for both discharges and higher order, others were only computed with a first order accuracy and SHQ for a fundamental comparison of the parameters influence. All cases with proper results are summarized in Table 5.3.

Table 5.3: *3D sensitivity study - Computed cases*

Mesh	Discharge [m^3/s]	Order	Nr. of Elements
Base mesh - symmetrical	SHQ, BHQ	1^{st}, 2^{nd}	711916
Base mesh - symmetrical - added roughness	SHQ	2^{nd}	711916
Base mesh - non symmetrical	SHQ	1^{st}	1403637
Fine mesh - symmetrical	SHQ	1^{st}	2198944

5.3.1 Computation time

As 3D computations take a long time, it was necessary to reduce the computed time steps to a minimum. This was done by observation of the water level at a certain point with the basic model.

The chosen location was right in the channel's mid at the symmetry plane and at the general measuring axis 45 metres upstream of the flaps rotation axis. As in the 2D simulations, two steps were performed regarding the convection term scheme. Again, the first step was computed with `Gauss upwind` and the second was changed to `Gauss linearUpwind grad(U)`. Linear upwind differencing in the second step led to a faster computation and was less sensitive to the initial state and therefore ensured a stable simulation. The results of this analysis for SHQ are given in Figure 5.5.

This graph shows that a first order computation of 70 seconds was sufficient to reach an acceptable initial state for the second run. A higher accuracy of the linear upwind differencing resulted in a slight increase of the water level. After approximately 30 seconds, so at a total time of 100 seconds, this again led to a stable state. The variation of measurements was about two centimetres and showed the common wave propagation of an open channel flow.

The 3D computation time comparison is given in Table 5.4. It was done for the first three cases from Table 5.3 for SHQ and first order computation. These values again depend on the used hardware given before in Table 4.6. This comparison showed, that non symmetrical cases would take a very long time for a complete result. Taking into account that models with remaining diversion dam would need a higher refinement, this would lead to an unreasonable long runtime with the available computational power. A simulation with an overall mesh refinement would take approximately 8 times longer compared to the basic mesh size, even if symmetry was used.

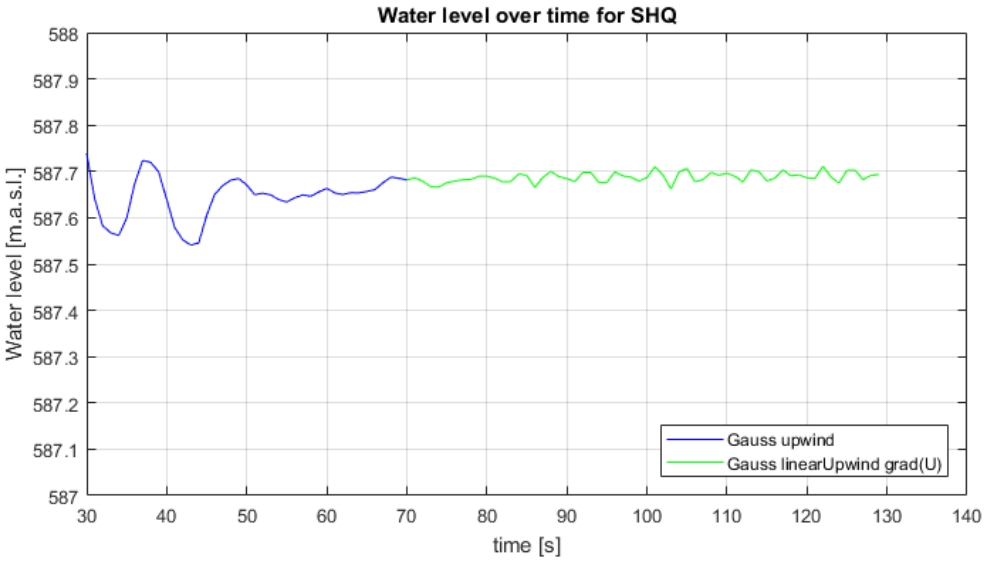


Figure 5.5: *Water level development over time for SHQ*

Table 5.4: *Computation time comparison in 3D*

Mesh	**Δt [s]**	**Iteration time [s]**	**$\frac{\text{Iterationtime}}{\Delta t}$**	**Normalized**
Base mesh - symmetrical	0.0031	1.30	425	0.126
Base mesh - non symmetrical	0.0025	2.65	1042	0.308
Fine mesh - symmetrical	0.0016	5.43	3378	1

5.3.2 Water Level / Overflow Coefficient

All results of this sensitivity study for the water level at the measuring point and their related μ_w values calculated with Poleni are given in Table 5.5. Due to the open channel wave propagation that can be seen in Figure 5.5, the reading accuracy decreased. In the 2D simulations, this transient behaviour was not showing to this extent.

These values showed that a non-symmetrical mesh did not result to a significant difference within the reading accuracy. Like in the 2D analyses, a mesh refinement led to a slight decrease of the water level for these element sizes, whereas an increase of the order led to a slight rise of water level. The only modification that caused a minor difference of μ_w was the addition of roughness. Nevertheless it was chosen to neglect this influence, as the physical model could be considered smooth.

Table 5.5: *3D sensitivity study - SHQ Comparison*

Case	H_{FO} [m.a.s.l.]	$\mu_{w,FO}$	H_{SO} [m.a.s.l.]	$\mu_{w,SO}$
Base mesh - symmetrical	587.68	0.75	587.69	0.75
Base mesh - symmetrical - added roughness	-	-	587.71	0.74
Base mesh - non-symmetrical	587.68	0.75	-	-
Fine mesh - symmetrical	587.66	0.75	-	-

5.4 Contraction Behaviour

As the main reason for the increase of water levels from 2D to 3D simulations was the contraction, the influence of this effect was investigated for SHQ. A comparison was done for the calculated 3D models with an elliptic and without any pillar modification for SHQ. First, the general behaviour in the pillar's area was visualized using streamline plots. Those are shown in Figures 5.6 and 5.7. It can be clearly seen that the contraction in a flume situation with a sharp pillar is higher than with an elliptic extension. The computation with a non-modified pillar showed a clear separation zone. This effect may decrease due to the widening in a natural reservoir.

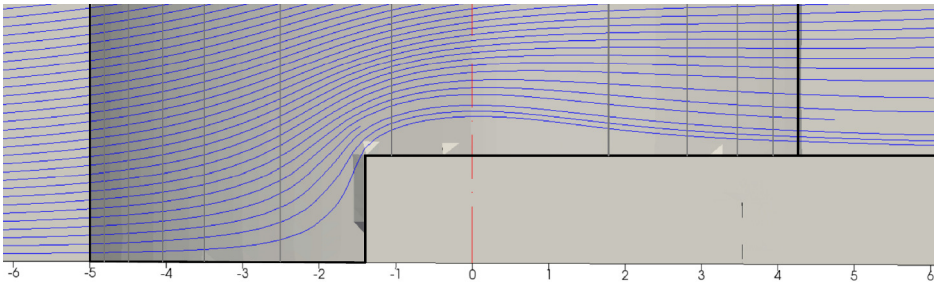


Figure 5.6: *Contraction behaviour - Streamlines without Pillar modification*

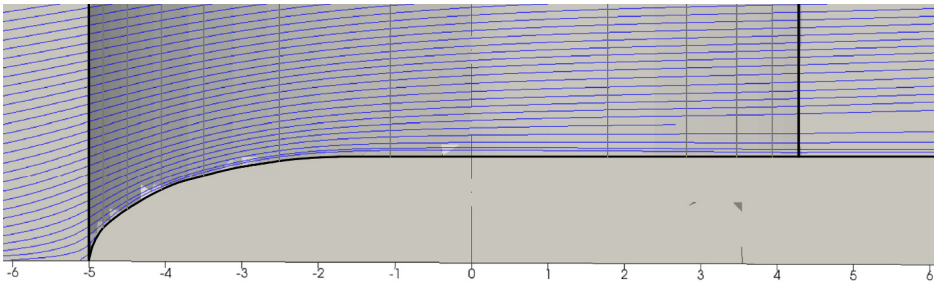


Figure 5.7: *Contraction behaviour - Streamlines with Pillar modification*

The acquired water levels at the measuring point and overflow coefficients of both geome-

tries with first and second order accuracy are given in Table 5.6. Just due to contraction the water level rose approximately by 0.1 metre, thus leading to a decrease in the overflow coefficient.

Table 5.6: *Pillar Modification - SHQ comparison*

Case	H_{FO} [m.a.s.l.]	$\mu_{w,FO}$	H_{SO} [m.a.s.l.]	$\mu_{w,SO}$
Base mesh - symmetrical	587.68	0.75	587.69	0.75
Modified base mesh - symmetrical - without pillar modification	587.78	0.73	587.79	0.73

As described in Section 2.2.2, the higher contraction does not show in nature, therefore the basic 3D model was used for the final comparison. This case offered the advantage of a reasonable computation time and represents the physical model tests state.

5.5 Turbulence Parameters

The turbulence parameters for the 3D computations were again calculated using the formulas from Versteeg and Malalasekera [2007] given in Section 3.3. Unlike in the 2D case given in Section 4.5, the hydraulic diameter was chosen for the characteristic length L. This parameter can be calculated with Equation 5.1. The hydraulic diameter consists of the cross section A and the wetted area P, so of the water level at the inlet H_{Inlet} and the width W. The symmetry was neglected for this.

$$D_H = \frac{4A}{P} = \frac{4H_{Inlet}W}{2H_{Inlet} + W} \tag{5.1}$$

Again the acquired values were validated with the experimental formulas using the wall shear stresses from prior 3D computations. The parameters decreased slightly in respect to the 2D values due to the full channel's width, but so did the shear stresses. This validation showed the same outcome as in the 2D comparison in Figure 4.11, therefore the values were not adjusted. The used input constants are given in Table 5.7.

Table 5.7: *Turbulence parameters for 3D analysis*

Load case	k	ϵ
BHQ	0.005053125	0.011340703
SHQ	0.00003605366	0.00011389057

5.6 Results

Finally full results for both discharges regarding water level and overflow coefficient are shown for the physical model test, 3D, and 2D simulations. These are furthermore compared to the desired values within this section. Moreover some plots of the pressure development parallel to the weir's axis are given.

5.6.1 Water Level / Overflow Coefficient

As for the 2D cases, water levels and overflow coefficients μ_w, determined with the Poleni formula, are compared here. The acquired data is given in Table 5.8 for both flows. The μ_w coefficients are furthermore displayed in Figure 5.8. Taking the velocity v_0 into account by using the Weisbach formula would again lead to a loss of efficiency for the model test and CFD simulations. The loss regarding μ_w would again be approximately 0.02-0.03 for BHQ and 0.03-0.04 for SHQ.

Table 5.8: μ_w *- Comparison of desired value, physical model test and CFD 2D/3D*

Case	$\mathbf{H_{BHQ}}$ **[m.a.s.l.]**	$\boldsymbol{\mu}_{\mathbf{w,BHQ}}$	$\mathbf{H_{SHQ}}$ **[m.a.s.l.]**	$\boldsymbol{\mu}_{\mathbf{w,SHQ}}$
Illwerke - Desired values	585.90	0.73	587.73	0.74
Physical - Without diversion dam	586.23	0.65	588.07	0.68
CFD 2D - Without diversion dam	585.92	0.72	587.56	0.77
CFD 3D - Without diversion dam	585.98	0.71	587.69	0.75

As expected, the water level had increased from 2D to the 3D simulations. Comparing values from the physical model test to the 3D computations showed an increase of approximately 9% for BHQ and 10% for SHQ regarding the overflow coefficient.

A comparison of the results regarding μ_w to literature, such as Rössert [1994] given in Section 2.2.3, showed a good efficiency for this weir's shape. Whereas the physical model tests results did not reach the desired values, the computational 3D results were quite close. For BHQ the water level exceeded the limits and for SHQ it was slightly below the desired state. Nevertheless, this small difference should be treat with caution. Note that a model without pillar modification would exceed the limits for both discharges. A comparison of the water level in respect to the desired height, as done for the physical model test in Table 2.5, is given in Table 5.9.

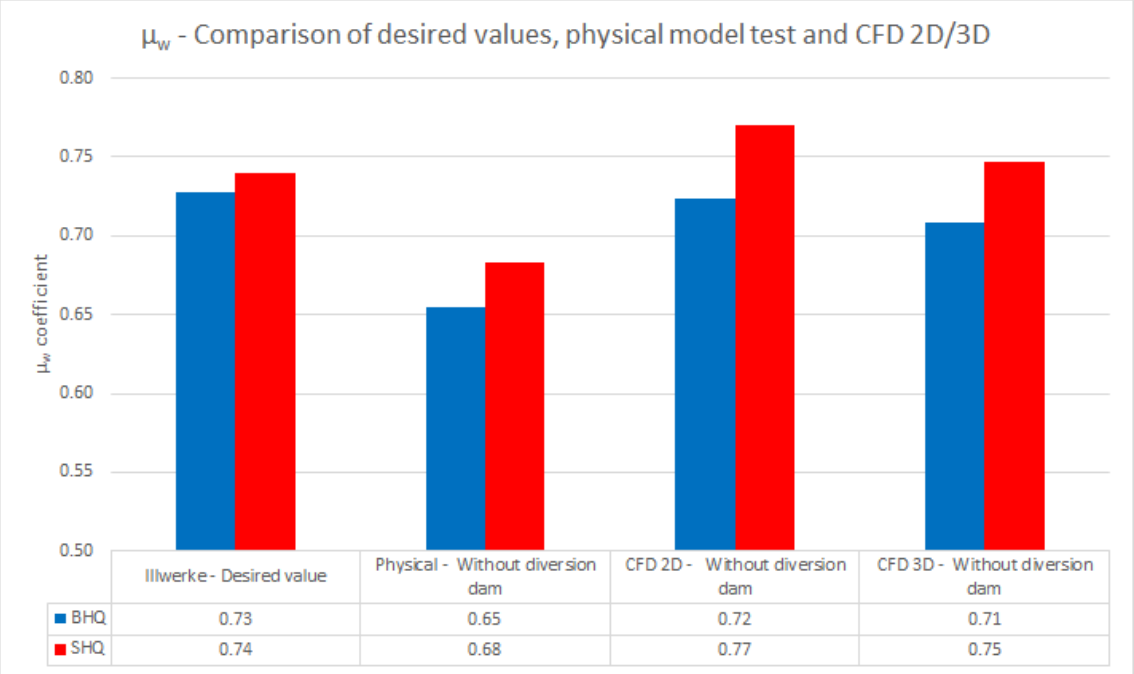


Figure 5.8: μ_w *- Comparison of desired values, physical model and CFD*

Table 5.9: *Comparison - Difference in water level 3D CFD and desired values*

Discharge	**Water level (Model - Desired value)**	**Deviation to desired value**
BHQ = 397 m^3/s	585.98 [m.a.s.l.] (Desired water level = 585.90 [m.a.s.l.])	+ 0.08 [m]
SHQ = 679 m^3/s	587.69 [m.a.s.l.] (Desired water level = 587.73 [m.a.s.l.])	- 0.04 [m]

The top level of the existing structure has an elevation of 588.10 [m.a.s.l.]. Whereas there was only a free board of three centimetres left in the physical model test, there was still a space of approximately 41 centimetres left in the 3D CFD simulation.

To finalize the water level comparison, the drawdown curves for physical model test and both CFD simulations are given in Figure 5.9. The water level is shown from the measuring point to the weir's tip, with 3D CFD results right next to the symmetry plane. An overall rise of water level from 2D to 3D can be clearly seen.

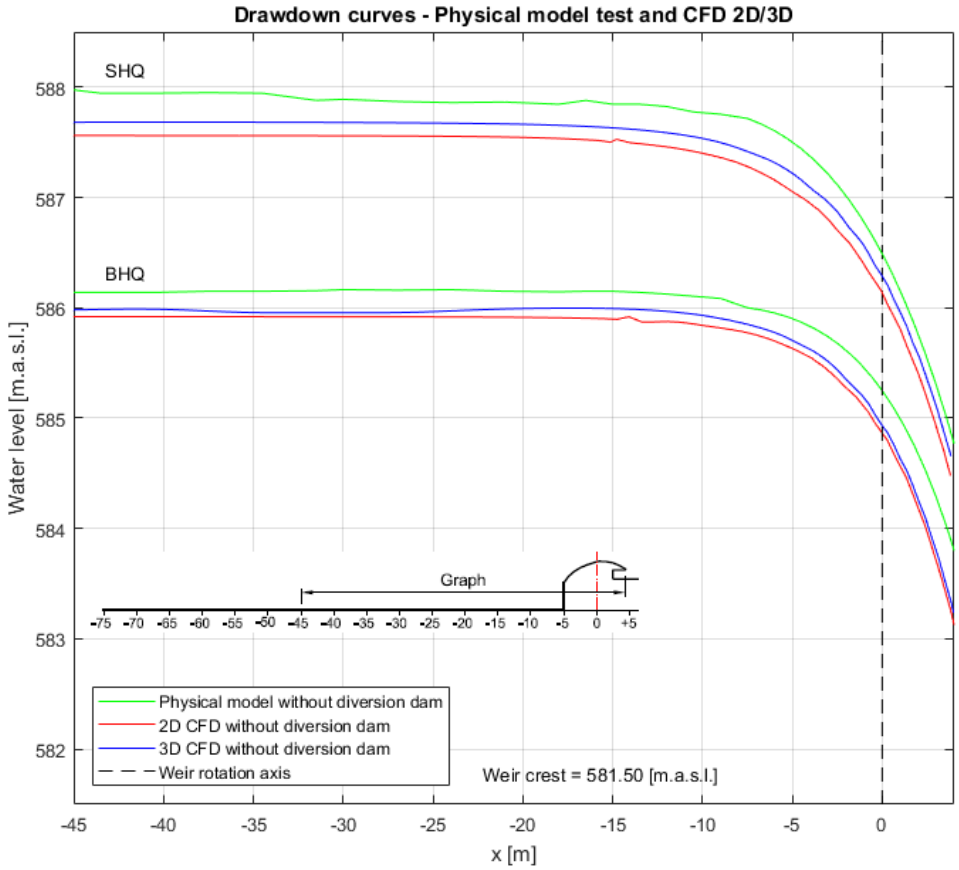


Figure 5.9: *Drawdown curves - Physical model test and CFD 2D/3D*

5.6.2 Pressure Distribution

The pressures varied to a certain extend over the weir's width. Development of pressures over the weir's width can be best seen in line plots across the y-axis. These plots were done at three remarkable locations, which are:

- Rotation axis of the flap
- Weir crest
- Tip of the flap

The plots are given in Figures 5.10 to 5.12. In these graphs the side walls location is approximately at a y-coordinate of 1.40 and the symmetry plane at 11.40 metres. As can be seen on those figures, the pressures drop right next to the wall and increase to the weir's mid. All three plots are located within the second half of the weir. Whereas SHQ pressures were higher in the first half, BHQ pressures exceeded them in the second one.

It has to be noted that due to the coarser mesh, the flap's tip is not snapped completely correct, therefore the location of this edge is slightly shifted from 2D to 3D. Furthermore, minor mesh uncertainties caused the pressure drop in Figure 5.12 right next to the symmetry plane. This drop did not show in a non-symmetrical case.

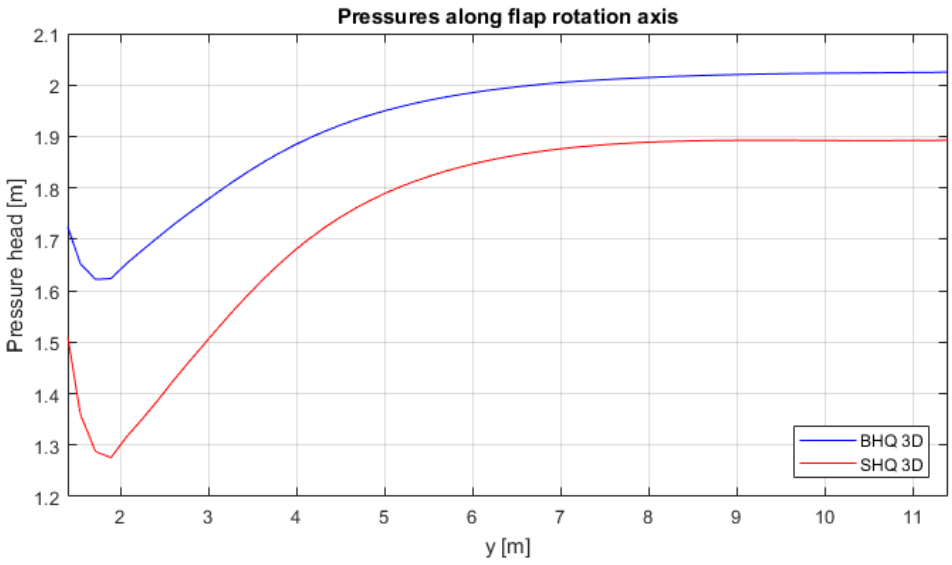


Figure 5.10: *Pressures in y-direction - Weir rotation axis*

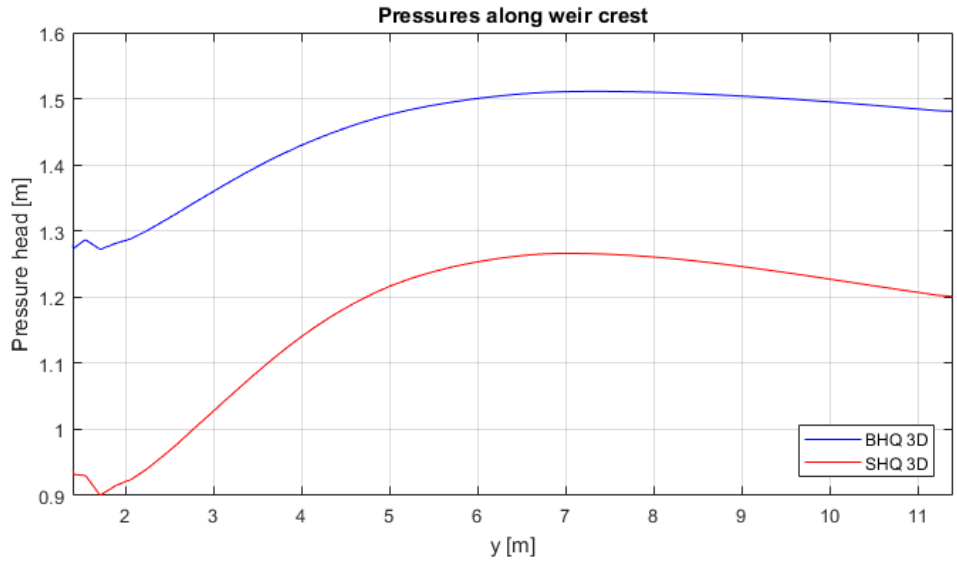


Figure 5.11: *Pressures in y-direction - Weir crest*

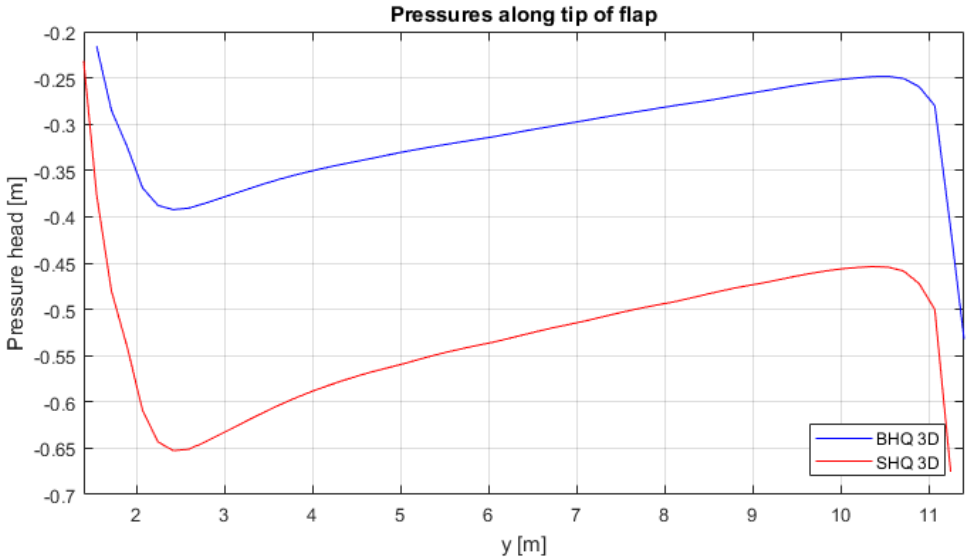


Figure 5.12: *Pressures in y-direction - Tip of the flap*

At a distance of approximately five metres from the side pillar pressures reached a mostly constant state. Due to the symmetry this means that a mid section of 10 metres had a constant pressure. For a comparison of 3D and 2D pressures, a cross section at a y-coordinate of 10 metres was chosen. This cross section represents the weir's mid, neglecting irregularities caused by mesh uncertainties. Comparing those results to the 2D data from Section 4.6.2 shows a general increase of pressures within the main field.

A comparison of 3D and 2D pressures and their resulting differences are given in Figures 5.13 and 5.14. The coarser mesh of the 3D simulations led to a less smooth result and furthermore to a high difference at the rotation axis. Clearly, the 2D computations under-

estimate the main fields pressures. In the three dimensional case, BHQ pressures were on average 0.1 metres and SHQ pressures on average 0.25 metres higher than their respective 2D results. Most of this difference is related to an increase of hydrostatic pressure due to a rise of water level over the weir. This is shown in Figure 5.9. The average increase of water level at the weir is approximately 0.08 metres for BHQ and 0.16 metres for SHQ. Despite a rising water level, 3D pressures could fall below their 2D results next to the wall, especially at the flap.

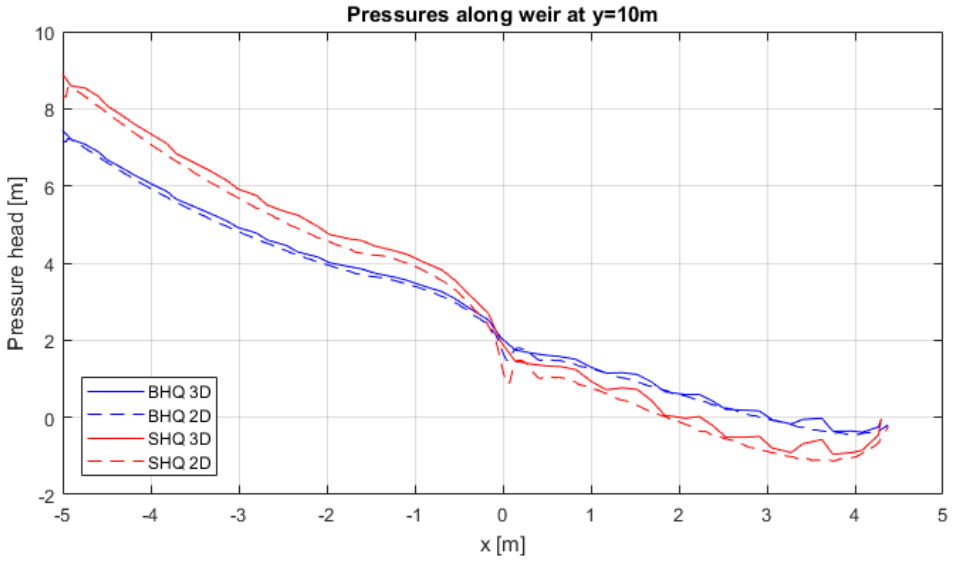


Figure 5.13: *Pressures in x-direction at y=10m - Comparison 3D/2D*

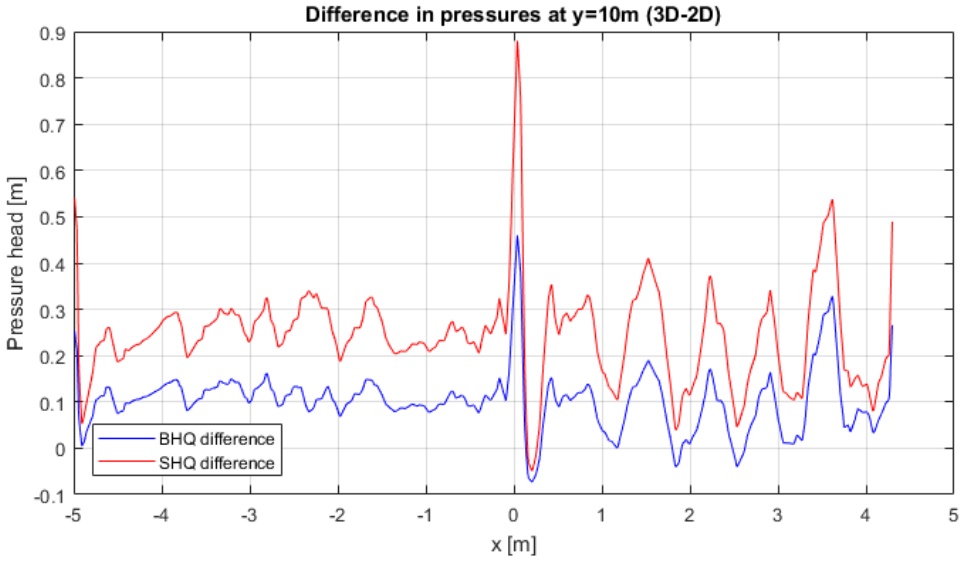


Figure 5.14: *Pressures in x-direction at y=10m - Difference 3D/2D*

5.7 Summary

Meshing in 3D showed a totally different behaviour compared to 2D regarding the two used tools. Whereas *Salome* had a lot of advantages in 2D, most of them vanished in 3D. The most common *Salome* algorithms prefers tetrahedrons. This leads to a rise of elements, a highly unstructured mesh, and unstable computations for coarse meshes. Nevertheless *Salome* offers a good opportunity to prepare the necessary STL files for snappyHexMesh. SnappyHexMesh worked well in the 3D cases. It created meshes with a good performance, a reasonable computation time, and a good structure within the domain. Its only disadvantage in 3D is the necessary refinement to snap edges correctly. If some minor changes in the geometry are accepted, coarser meshes are still possible.

The computation time analysis showed that only symmetrical models with a rather coarse mesh could be computed within a reasonable time. Therefore only cases without diversion dam were simulated as only those could be used with a symmetry boundary condition.

In the sensitivity study done for the 3D simulations, the examined variations showed the same behaviour as in 2D. The basic mesh size was sufficient to examine this weir's overflow with an acceptable accuracy. Other factors such as symmetry boundary condition and roughness did not show a noteworthy difference. Only a geometrical change with a sharp side pillar form led to a significant increase of the water level in this flume situation. As the most natural state was reached with an elliptic modification in the physical model test, this geometry was used for final comparison.

The final comparison of water heights and the overflow coefficients μ_w showed differences between the 3D CFD results and the physical model test. Whereas the data acquired from the model test did not meet the operator's desired state, the three dimensional numerical simulations were quite close to it. Nevertheless, those result have to be treated with caution as the used geometry represents a flume flow as in the model test.

The pressure evaluation showed that the three dimensional influence cannot be neglected. Pressures parallel to the weir's axis were lower next to the weir's side and increased to the weir's mid. An overall comparison of pressures along the weir showed a general increase in the three dimensional simulation. This growth is mainly caused by an increased water level in this area leading to a higher hydrostatic pressure, but also by a change of velocity and therefore a different hydrodynamic pressure.

As all observed parameters showed the same behaviour in 2D and 3D, there is the possibility to recreate 3D results by use of a 2D model, using for example a higher discharge to reach the 3D state.

6 Aeration

6.1 Preliminaries

An adequate ventilation is necessary to ensure a free overflow and prevent changing pressures that would cause a dynamic load on the flap. Due to a removal of the mid pillar, the existing aeration area will decrease from 2.16m^2 to 1.25m^2. As mentioned in Section 2.2.3, the physical model test showed signs of insufficient aeration for SHQ.

Empirical formulas regarding aeration are referring to the necessary area of the ventilation shafts. Some of these formulas are listed in Table 6.1. In the model test's report (IWB [2014]), the criteria of Strobl and Zunic [2006] given in Equation 6.1 was used. This approach led to a necessary ventilation area of approximately 2.00m^2. CFD analyses give a different opportunity to estimate the necessary aeration. Figure 6.1 shows an aerated 2D model under SHQ operation.

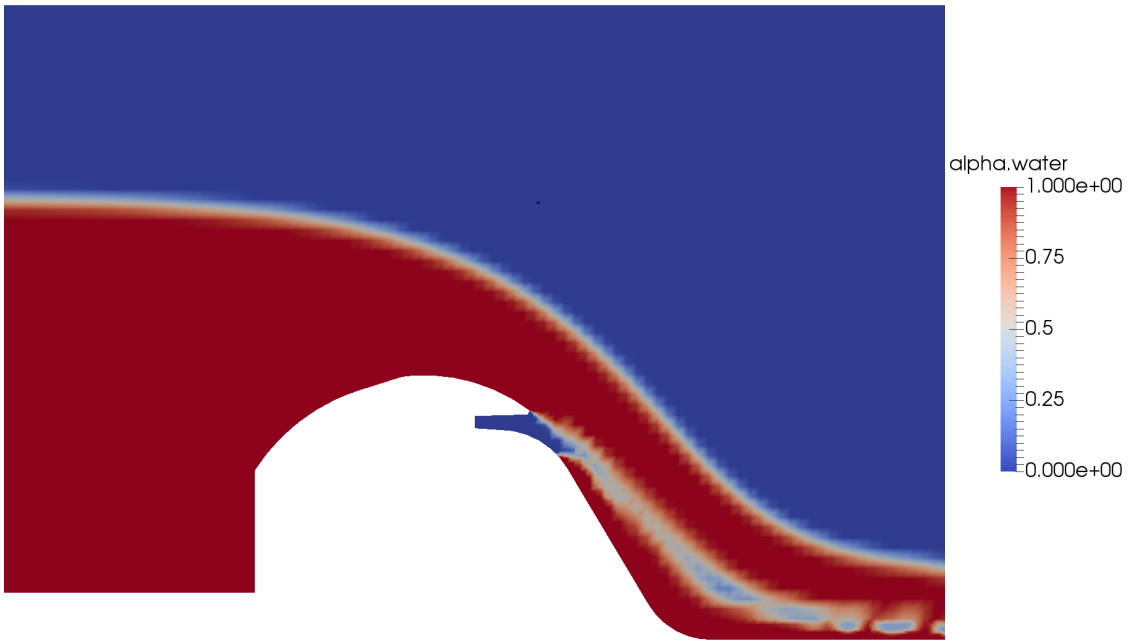


Figure 6.1: *Aeration in a CFD model*

Table 6.1: *Formulas for estimation of the necessary aeration area*

Formula	**Variables**
Strobl and Zunic [2006] $A = 0.015 \cdot W \cdot z$ (6.1)	• A = Aeration area [m^2] • W = Weir's width [m] • z = Height difference from lowered flap to downstream water level [m]
Li [2014] $D = 0.11 \cdot H_{max} \cdot W^{0.5}$ (6.2)	• D = Diameter of aeration pipe [m] • W = Weir's width [m] • H_{max} = Biggest water head occurring at the weir [m]
Raikar [2012] $A = 0.5\%(WZ)$ (6.3)	• A = Aeration area [m^2] • W = Weir's width [m] • Z = Downstream water depth [m]

6.2 CFD analysis

To create a model including aeration, such as in Figure 6.1, the geometry was extended to the stilling basin. This model enlargement leads to an increase of elements and an area with high velocities at the spillway. Whereas an increase of elements leads to a general rise of computation time, higher velocities lead to an increase of the Courant number which decreases the time step. To prevent high computation times, this analysis was performed in 2D. For the aeration, a patch field below the flap was created. This field used the same boundary condition as the models surface, allowing air inflow. Whereas the water level upstream of the weir remained mostly stable after enough time steps, the downstream part behaved highly transient but periodic.

First attempts for an estimation of the required ventilation area with CFD were modelled with a geometrical reduction of the aeration patch. This led to a very small air inlet, which produced massive velocities in this area and a very slow computation due to the Currant number. Nevertheless, it was not possible to create a simulation showing an insufficient aeration, as the air inflow behaved non-physically. Therefore it was decided to give the necessary amount of air by use of the airflow.

The necessary airflow is acquired by using *OpenFOAM's* `flowRatePatch` post-processing function, which gives the flow of a patch in m^3/s. A total of 30 time steps, with an already stable upstream situation, were examined. In Figure 6.2, the results of each step are plotted for both discharges. Their minimum, maximum, and mean values are given in Table 6.2. The results are already scaled up to the total weir's width.

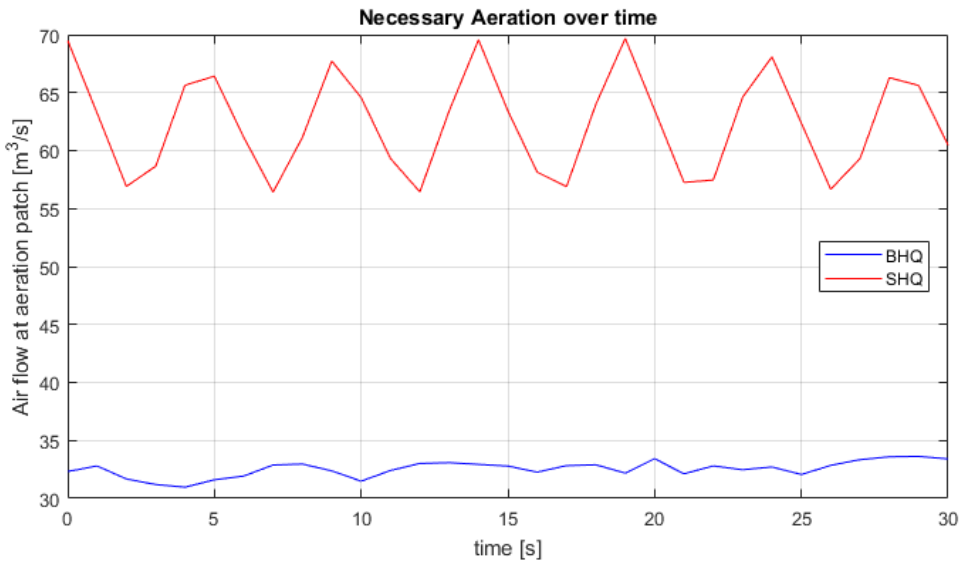


Figure 6.2: *Aeration - Flow Rate over Time*

Table 6.2: *Aeration - Airflow rate minimum, maximum and mean*

Discharge	**Minimum [m^3/s]**	**Maximum [m^3/s]**	**Mean [m^3/s]**
BHQ	30.97	33.62	32.55
SHQ	56.42	69.67	62.38

As expected, SHQ is decisive regarding aeration. The necessary air amount has almost doubled. All of the empirical formulas listed in Table 6.1 estimate the necessary area, therefore a direct comparison with the acquired CFD results is not possible. Nevertheless, since the criteria of Strobl and Zunic [2006] was used in the model tests report (IWB [2014]) some attention to this formula from Equation 6.1 was paid. According to this criteria, the ventilation area depends on the flaps elevation and the downstream water level. As the smaller discharge of BHQ obviously has a lower water level, the required aeration area would increase. Nevertheless, the CFD simulation clearly leads to a higher air inflow with increased discharge. Therefore, the aeration area should increase with a higher flow.

7 Conclusion

All computations within this theses were conducted with *OpenFOAM*. For pre- and post-processing other non commercial tools were used. The findings in this thesis can be divided into several parts. Two different meshing tools were used for preprocessing, allowing a direct comparison of the used meshing algorithms. The requirements to the meshes were a high accuracy with a reasonable computation time. To be able to estimate the error of computations, a sensitivity study regarding the mesh size was performed. In addition, also studies regarding roughness and turbulence input parameters were conducted. As in the physical model test, different geometries were used for comparison. Noteworthy are models with and without the diversion dam, that is still existing in the reservoir.
To allow a comparison of different models, the overflow coefficient μ_w was used. In addition, pressures occurring at the weir's surface were analysed. Due to an insufficient aeration in the model test, a CFD analysis to investigate the required aeration of this particular weir was performed.

For preprocessing the external program *Salome* and *OpenFOAM's* own meshing algorithm snappyHexMesh were used. Both tools showed differences in 2D and 3D. Their advantages and disadvantages for 2D are summarized in Table 4.2.
Salome has a clear advantage due to its graphical user interface. Furthermore, it has no restrictions regarding the geometry. *Salome's* main disadvantage is its tetrahedral preference. Tetrahedrons lead to an unstructured mesh with convergence problems in *OpenFOAM*.
OpenFOAM's snappyHexMesh is quite error prone, nevertheless it is a good alternative for creation of hexahedral meshes. It generates well structured meshes with a good performance in *OpenFOAM*. The main disadvantage of this tool is the inevitable creation of geometrical inappropriate elements, which lead to a slight change in more complicated geometries. This effect can be reduced with a refinement along these surfaces, leading to an increase of the element number. For computations within this theses only meshes created with snappyHexMesh were used.

The sensitivity study showed the same behaviour in 2D and 3D analyses with respect to observed parameters. The performance of a mesh sensitivity study in 2D showed that even with rather coarse meshes a sufficient accuracy regarding the water level and its related μ_w coefficient can be achieved. A massive rise of computation time with finer meshes led to the choice of using coarser meshes, as those had a reasonable computation time even in 3D.

Adding roughness to the models wall boundaries showed a minor rise of the water level. As this change is rather small and the physical model test could be considered smooth, this difference was neglected.
As described in Section 4.5, the use of a simple formula from Versteeg and Malalasekera [2007] for the turbulence input parameters k and ϵ is sufficient in such simulations and it is not necessary to use more complex experimental formulas.

As in the physical model test, a removal of the diversion led to a decrease of the water level in the reservoir, thus to an increase of the weir's efficiency. This effect can be seen in Section 4.6.1 including changes of the draw-down curves. Whereas the dam's effect on the reservoir did not show for BHQ, it had a huge impact on the results of a SHQ analysis. As a 3D simulation regarding a removal of the diversion dam would need a full scale model without symmetry, this effect was not tested in a three dimensional case. It has to be kept in mind that the dam's alignment is not parallel to the weir. This may increase the dam's impact in 3D CFD on the observed parameters.

The main task was to compare the water levels at a specific measuring point and their related μ_w coefficient. In a final comparison only models without diversion dam and an elliptical pillar modification were used. This comparison is given in Section 5.6.1. The results showed that the water levels received from CFD analysis were different to those obtained from the physical model test. Numerical analyses led to significantly lower water levels compared to the model test. 3D computations led to a decrease of the overflow coefficients compared to 2D simulations due to contraction, but this effect had a smaller impact than expected. Compared to related literature, the examined weir had a high efficiency.
Results of the numerical analyses were quite close to the operator's desired values. As the computational model represents a flume setup, like the physical model test, a change of these values is possible when a more natural geometry will be used. Furthermore, it has to be noted that for the main comparison of the weir's efficiency only the Poleni formula was used. Despite velocities of $> 1.5m/s$ in the flume this formula was used instead of Weisbach, as it was used in the physical model test's report. Including the velocities would lead to a significant decrease of μ_w shown in Section 5.6.1.

The pressures obtained from 2D analyses given in Section 4.6.2 showed an increase of their respective minima and maxima with higher discharges. In general, pressures were positive in the first part of the weir, reaching negative values at the flap. Furthermore, it can be seen that a removal of the dam would lead to a decrease of negative pressures.
3D simulations were revealing a change of pressures over the weir's width. Pressures parallel to the weir's axis showed a drop at the weir's side wall. After increasing up to a distance of approximately five metres from the sides, pressures reached a mostly constant state. Compared to the 2D computations, the pressures acquired from 3D simulations at the weir's mid field were higher. The main reason for this was the increased hydrostatic pressure due to a rise of water level. Furthermore, there was also a slight change of the dynamic pressures as velocities changed. The comparison of 2D and 3D simulations showed that a first estimation using 2D results can be quite accurate, when keeping these changes in mind.

There are several different formulas available regarding aeration of weirs. All of these formulas empirically estimate the necessary cross section of aeration shafts. Whereas these formulas are estimating the necessary area, CFD gives another opportunity. Numerical multiphase models allow to acquire the necessary air flow. The result of this analysis is given in Section 6.2 for both load cases. Finding a relation between the acquired airflow and the necessary aeration area would need further research.

Finally, the main points within this conclusion are summarized as follows:

- In 2D, hexahedral mesh generation with *Salome* is quick and simple. On the other hand, snappyHexMesh performed better in 3D.
- Coarser meshes are sufficient for quick approximation of water levels and discharge coefficients
- CFD vs. physical model test
  - Water level: In all cases CFD results were lower than in the physical model test
  - Subsequently the recalculated overflow coefficients are higher
- The results from 2D and 3D simulations (water levels, pressures) showed only minor differences
- Aeration formulas from literatures are only referring on the necessary area, but CFD opens new possibilities
- It is possible to acquire all these results with open-source software

8 Outlook

This thesis covers investigations regarding water levels, overflow coefficients, pressures, and aeration, but with the capabilities of CFD also some extra goals are achievable. Since physical model tests are scaled, there will always be a certain error due to slight movements of the water levels and inaccuracies of measuring devices. As CFD allows full scale modelling, there would be the possibility to compare results with the prototype in a different way. Therefore, validating CFD computations with measurements in nature is recommended.

Further research topics regarding this type of weir are given below.

- Comparing CFD with the prototype
  - Creation of a CFD model using a natural geometry to investigate the influence of e.g. contraction
  - Combination of measurements in the prototype regarding the airflow in the aeration shafts with CFD , allowing to create an empirical formula for aeration based on the airflow
- Estimation of errors occurring in physical model tests and CFD

Bibliography

Santiago Márquez Damian. Description and utilization of interfoam multiphase solver. Technical report, International Center for Computational Methods in Engineering, 2012.

D Gisen. Generation of a 3D Mesh Using snappyHexMesh Featuring Anisotropic Refinement and Near-wall Layers. In *Proceedings of the 11th International Conference on Hydroscience and Engineering (ICHE), Hamburg, Germany*, pages 983–990, 2014.

Christopher J Greenshields. OpenFOAM 2.4. 0 Programmers Guide. *OpenFOAM Foundation Ltd*, 2015.

Christopher J Greenshields. OpenFOAM 5.0 User Guide. *OpenFOAM Foundation Ltd*, 5 (1), 2017.

Fabian Gottfried Heinzle. 3D-Numerische Berechnung der Wehrförderfähigkeit für ein Flusskraftwerk. Master's thesis, Institute of Hydraulic Engineering and Water Resources Management, Graz University of Technology, 2014.

Cyril W Hirt and Billy D Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981.

DKH Ho, KM Boyes, and Shane M Donohoo. Investigation of spillway behavior under increased maximum flood by computational fluid dynamics technique. In *14th Australasian Fluid Mechanics Conference, Adelaide University, Adelaide, Australia*, 2001.

IWB. Hydraulischer Modellversuch, Speicher Gstins. Technical report, Institute of Hydraulic Engineering and Water Resources Management, Graz University of Technology, Juli 2014. Internal documents.

Hrvoje Jasak. *Error analysis and estimation for finite volume method with applications to fluid flow*. PhD thesis, Imperial College of Science, Technology and Medicine, London, 1996.

O. Kirschmer. *Tabellen zur Berechnung von Entwässerungs-Leitungen nach Prandtl-Colebrook*. Lüdecke, 1974.

Juha Kortelainen. Meshing tools for open source CFD-a practical point of view. *VTT, Espoo, Finland, Tech. Rep*, 2009.

Yan Li. Prediction of flow over thin-plate rectangular weir. In *Hydraulic Engineering III: Proceedings of the 3rd Technical Conference on Hydraulic Engineering (CHE 2014), Hong Kong, 13-14 December 2014*, pages 79–83. CRC Press, 2014.

Iehisa Nezu. *Turbulence in open-channel flows*. A.A. Balkema, Rotterdam, Brookfield, 1993.

Marcela Politano, Yushi Wang, Troy Lyons, and Larry Weber. Computation of the Flow Field over Spillways using OpenFoam. *EWRI World Environmental & Water Resources Congress*, 2016.

R.V. Raikar. *Laboratory Manual Hydraulics and Hydraulic Machines*. PHI Learning, 2012.

Robert Rössert. *Hydraulik im Wasserbau*. Oldenbourg, München, Wien, 1994.

Salome. Meshing documentation. http://docs.salome-platform.org/latest/gui/SMESH/index.html, 2017. Accessed: 18.08.2017.

L Schulze and C Thorenz. The multiphase capabilities of the cfd toolbox openfoam for hydraulic engineering applications. In *Proceedings of the 11th International Conference on Hydroscience and Engineering (ICHE), Hamburg, Germany*, pages 1007–1014, 2014.

Theodor Strobl and Franz Zunic. *Wasserbau*. Springer, Berlin, Heidelberg, 2006.

Bram Van Leer. Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *Journal of computational physics*, 14(4):361–370, 1974.

K. H. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics*. Pearson Educcation Limited, Harlow, 2 edition, 2007.

VoGIS. Vorarlberg online Atlas. http://vogis.cnv.at/atlas/, 2017. Accessed: 01.07.2017.

VORARLBERG ONLINE. VKW Kraftwerk Lutz. http://www.vol.at/2008/04/VKW_Kraftwerk_Lutz.pdf, 2008. Accessed: 23.08.2017.

Vorarlberger Kraftwerke AG. Model request 2013. Technical report, April 2013. Internal documents.

Vorarlberger Kraftwerke AG. Kraftwerk Unterstufe Lutz. https://www.vkw.at/kw-unterstufe-lutz-unternehmen.htm, 2017. Accessed: 10.07.2017.

List of Figures

List of Tables

Appendix

8.1 Input File - *OpenFoam*

8.1.1 Folder 0

8.1.1.1 alpha.water

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.1                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
version     2.0;
format      ascii;
class       volScalarField;
location    "0";
object      alpha.water;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#include        "include/initialConditions"

dimensions      [0 0 0 0 0 0 0];
internalField   uniform 0;

boundaryField
{
inlet
{
type            variableHeightFlowRate;
lowerBound      0.3;
upperBound      0.95;
value           uniform 0;
}
```

```
outlet
{
type            zeroGradient;
}

surface
{
type            inletOutlet;
inletValue      uniform 0;
value           uniform 0;
}

bottom
{
type            zeroGradient;
}

weir
{
type            zeroGradient;
}

ffmaxy
{
type  empty;
}

ffminy
{
type  empty;
}

}


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

8.1.1.2 Turbulent dissipation rate

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  plus                                  |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
version     2.0;
format      ascii;
class       volScalarField;
location    "0";
object      epsilon;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#include        "include/initialConditions"

dimensions      [0 2 -3 0 0 0 0];

internalField   uniform $turbulentEpsilon;

boundaryField
{
inlet
{
type            fixedValue;
value           $internalField;
}

outlet
{
type            inletOutlet;
inletValue      $internalField;
value           $internalField;
}

bottom
{
type            epsilonWallFunction;
value           $internalField;
}

weir
{
```

```
type            epsilonWallFunction;
value           $internalField;
}


surface
{
type            inletOutlet;
inletValue      $internalField;
value           $internalField;
}

ffmaxy
{
type empty;
}

ffminy
{
type empty;
}

}


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

8.1.1.3 Turbulent kinetic energy

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  plus                                  |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
version     2.0;
format      ascii;
class       volScalarField;
location    "0";
object      k;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#include        "include/initialConditions"
```

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

internalField   uniform $turbulentKE;

boundaryField
{
inlet
{
type            fixedValue;
value           $internalField;
}

outlet
{
type            inletOutlet;
inletValue      $internalField;
value           $internalField;
}

bottom
{
type            kqRWallFunction;
value           $internalField;
}

weir
{
type            kqRWallFunction;
value           $internalField;
}


surface
{
type            inletOutlet;
inletValue      $internalField;
value           $internalField;
}

ffminy
{
type empty;
}

ffmaxy
{
type empty;
```

```
}

}


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

8.1.1.4 Turbulent viscosity

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

dimensions      [0 2 -1 0 0 0 0];

internalField   uniform 0;

boundaryField
{
inlet
{
type            calculated;
value           uniform 0;
}

outlet
{
type            calculated;
value           uniform 0;
}

bottom
{
```

```
type            nutkWallFunction;
value           uniform 0;
}

weir
{
type            nutkWallFunction;
value           uniform 0;
}

surface
{
type            calculated;
value           uniform 0;
}

ffmaxy
{
type  empty;
}

ffminy
{
type  empty;
}

}

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

8.1.1.5 Pressure

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

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

dimensions      [1 -1 -2 0 0 0 0];

internalField   uniform 0;

boundaryField
{
inlet
{
type            zeroGradient;
}

outlet
{
type            fixedValue;
value uniform 0;
}

bottom
{
type            fixedFluxPressure;
}

weir
{
type            fixedFluxPressure;
}


surface
{
type            totalPressure;
p0              uniform 0;
}

ffmaxy
{
type empty;
}

ffminy
{
type empty;
}

}
```

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

8.1.1.6 Velocity

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

#include        "include/initialConditions"

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

internalField   uniform (0 0 0);

boundaryField
{
inlet
{
type            variableHeightFlowRateInletVelocity;
flowRate        $inletFlowRate;
alpha           alpha.water;
value           uniform ($vxInital 0 0);
}

outlet
{
type            inletOutlet;
inletValue      uniform (0 0 0);
value           uniform (0 0 0);
}

bottom
{
type            noSlip;
}
```

```
weir
{
type            noSlip;
}

surface
{
type            pressureInletOutletVelocity;
value           uniform (0 0 0);
}

ffmaxy
{
type empty;
}
ffminy
{
type empty;
}

}

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

8.1.2 Folder constant

8.1.2.1 Transport properties

```
/*--------------------------------*- C++ -*----------------------------------*\
|       o          |                                                 |
|    o     o       | HELYX-OS                                         |
|   o   O   o      | Version: v2.4.0                                  |
|    o     o       | Web:     http://www.engys.com                    |
|       o          |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
version 2.0;
format ascii;
class dictionary;
location constant;
object transportProperties;
}

phases (water air);
water
{
```

```
materialName water;
transportModel Newtonian;
NewtonianCoeffs
{
}

rho rho [1 -3 0 0 0 0 0 ] 998.2;
mu mu [1 -1 -1 0 0 0 0 ] 0.001002;
nu nu [0 2 -1 0 0 0 0 ] 1.0038068523342016E-6;
Cp Cp [0 2 -2 -1 0 0 0 ] 4187.0;
Prt Prt [0 0 0 0 0 0 0 ] 0.9;
lambda lambda [1 1 -3 -1 0 0 0 ] 0.5985;
pRef pRef [1 -1 -2 0 0 0 0 ] 101325.0;
TRef TRef [0 0 0 1 0 0 0 ] 293.0;
beta beta [0 0 0 -1 0 0 0 ] 2.07E-4;
Pr Pr [0 0 0 0 0 0 0 ] 0.9;
Cp0 Cp0 [0 0 0 1 0 0 0 ] 4187.0;
rhoCp0 998.2;
}

air
{
materialName air;
transportModel Newtonian;
NewtonianCoeffs
{
}

rho rho [1 -3 0 0 0 0 0 ] 1.205;
mu mu [1 -1 -1 0 0 0 0 ] 1.9137E-5;
nu nu [0 2 -1 0 0 0 0 ] 1.5881327800829875E-5;
Cp Cp [0 2 -2 -1 0 0 0 ] 1006.0;
Prt Prt [0 0 0 0 0 0 0 ] 0.85;
lambda lambda [1 1 -3 -1 0 0 0 ] 0.024;
pRef pRef [1 -1 -2 0 0 0 0 ] 101325.0;
TRef TRef [0 0 0 1 0 0 0 ] 300.0;
beta beta [0 0 0 -1 0 0 0 ] 0.00333;
Pr Pr [0 0 0 0 0 0 0 ] 0.9;
Cp0 Cp0 [0 0 0 1 0 0 0 ] 1006.0;
rhoCp0 1.205;
}

sigma sigma [1 0 -2 0 0 0 0 ] 0.0;
```

8.1.2.2 Turbulence properties

```
/*--------------------------------*- C++ -*----------------------------------*\
|       o          |                                                 |
|    o     o       | HELYX-OS                                         |
|   o   O   o      | Version: v2.4.0                                  |
|    o     o       | Web:     http://www.engys.com                    |
|       o          |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
version 2.0;
format ascii;
class dictionary;
location constant;
object turbulenceProperties;
}

simulationType RAS;
RAS
{
RASModel kEpsilon;
turbulence on;
printCoeffs on;
kEpsilonCoeffs
{
label "Standard high-Re k-\u03B5";
fieldMaps
{
k k;
epsilon epsilon;
nut nut;
}

Cmu 0.09;
C1 1.44;
C2 1.92;
alphaEps 0.76923;
}

}
```

8.1.3 Folder system

8.1.3.1 Schemes

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

ddtSchemes
{
default         Euler;
}

gradSchemes
{
default         Gauss linear;
}

divSchemes
{
//div(rhoPhi,U)  Gauss linearUpwind grad(U); //used for SO 3D simulations
//div(rhoPhi,U)  Gauss linear;  //used for SO 2D simulations
div(rhoPhi,U)  Gauss upwind;
div(phi,alpha)  Gauss vanLeer;
div(phirb,alpha) Gauss linear;
div(phi,k)      Gauss upwind;
div(phi,epsilon) Gauss upwind;
div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
default         Gauss linear corrected;
}
```

```
interpolationSchemes
{
default         linear;
}

snGradSchemes
{
default         corrected;
}

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

8.1.3.2 snappyHexMesh dictionary

```
/*--------------------------------*- C++ -*----------------------------------*\
|       o          |                                                        |
|    o     o       | HELYX-OS                                                |
|   o   O   o      | Version: v2.4.0                                         |
|    o     o       | Web:     http://www.engys.com                           |
|       o          |                                                        |
\*---------------------------------------------------------------------------*/
FoamFile
{
version 2.0;
format ascii;
class dictionary;
location system;
object snappyHexMeshDict;
}

castellatedMesh true;
snap true;
addLayers true;
geometry
{
bottom.stl
{
type triSurfaceMesh;
name bottom;
appendRegionName false;
}

inlet.stl
{
type triSurfaceMesh;
name inlet;
appendRegionName false;
```

```
}

outlet.stl
{
type triSurfaceMesh;
name outlet;
appendRegionName false;
}

sides.stl
{
type triSurfaceMesh;
name sides;
appendRegionName false;
}

surface.stl
{
type triSurfaceMesh;
name surface;
appendRegionName false;
}

weir.stl
{
type triSurfaceMesh;
name weir;
appendRegionName false;
}

}

castellatedMeshControls
{
features
(
);
refinementSurfaces
{
bottom
{
level ( 0 0);
}

inlet
{
level ( 0 0);
}
```

```
outlet
{
level ( 0 0 );
}

sides
{
level ( 0 0);
}

surface
{
level (0 0 );
}

weir
{
level (1 1 );
}

}

refinementRegions
{
}

locationInMesh ( -45.0 11.399999618530272 10.0 );
maxLocalCells 100000;
maxGlobalCells 40000000;
minRefinementCells 0;
nCellsBetweenLevels 2;
resolveFeatureAngle 30.0;
allowFreeStandingZoneFaces true;
planarAngle 30.0;
maxLoadUnbalance 0.1;
}

snapControls
{
nSolveIter 100;
nSmoothPatch 3;
tolerance 4.0;
nRelaxIter 5;
nFeatureSnapIter 30;
implicitFeatureSnap true;
explicitFeatureSnap false;
multiRegionFeatureSnap false;
```

```
}

addLayersControls
{
layers
{
bottom
{
nSurfaceLayers 2;
}

sides
{
nSurfaceLayers 2;
}

weir
{
nSurfaceLayers 2;
}

}

relativeSizes true;
expansionRatio 1.0;
finalLayerThickness 0.3;
minThickness 0.1;
nGrow 0;
featureAngle 270.0;
slipFeatureAngle 70.0;
nRelaxIter 5;
nSmoothSurfaceNormals 1;
nSmoothNormals 3;
nSmoothThickness 10;
maxFaceThicknessRatio 0.5;
maxThicknessToMedialRatio 0.3;
minMedialAxisAngle 90;
nBufferCellsNoExtrude 0;
nLayerIter 50;
nRelaxedIter 20;
writeVTK false;
noErrors false;
layerRecovery 1;
growZoneLayers false;
projectGrownUp 0.0;
}

meshQualityControls
```

```
{
maxNonOrtho 65.0;
maxBoundarySkewness 10;
maxInternalSkewness 4;
maxConcave 80.0;
minFlatness 0.5;
minVol 1.0E-13;
minTetQuality 1.0E-50;
minArea -1.0;
minTwist 0.02;
minDeterminant 0.001;
minFaceWeight 0.05;
minVolRatio 0.01;
minTriangleTwist -1.0;
nSmoothScale 4;
errorReduction 0.75;
relaxed
{
maxNonOrtho 75;
}

}

debug 0;
mergeTolerance 1.0E-6;
autoBlockMesh false;
```

8.2 Existing weir structure / Renovation

8.2.1 Plans

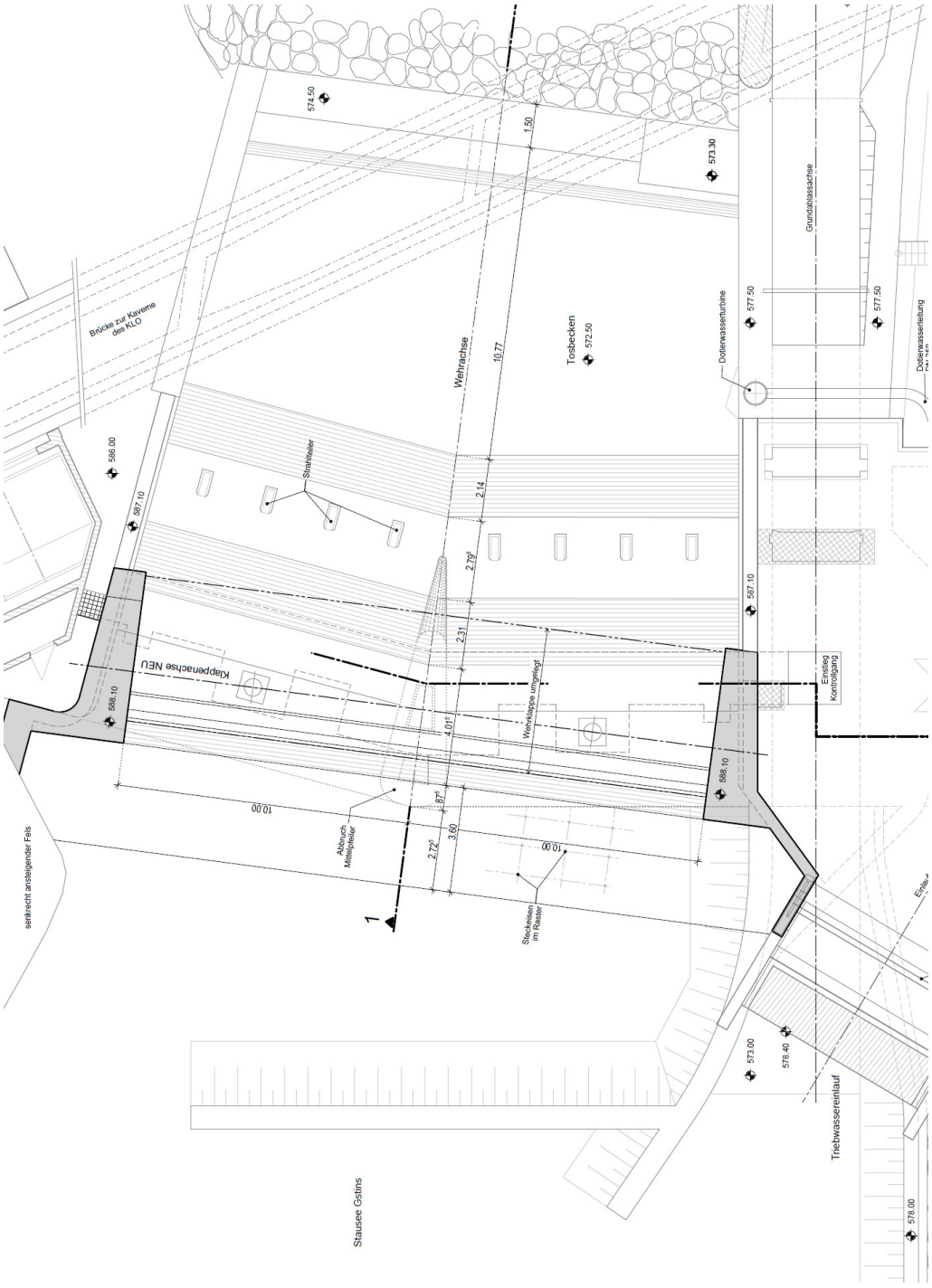


Figure 8.1: *Renovation ground view - Full structure - Vorarlberger Kraftwerke AG [2013]*

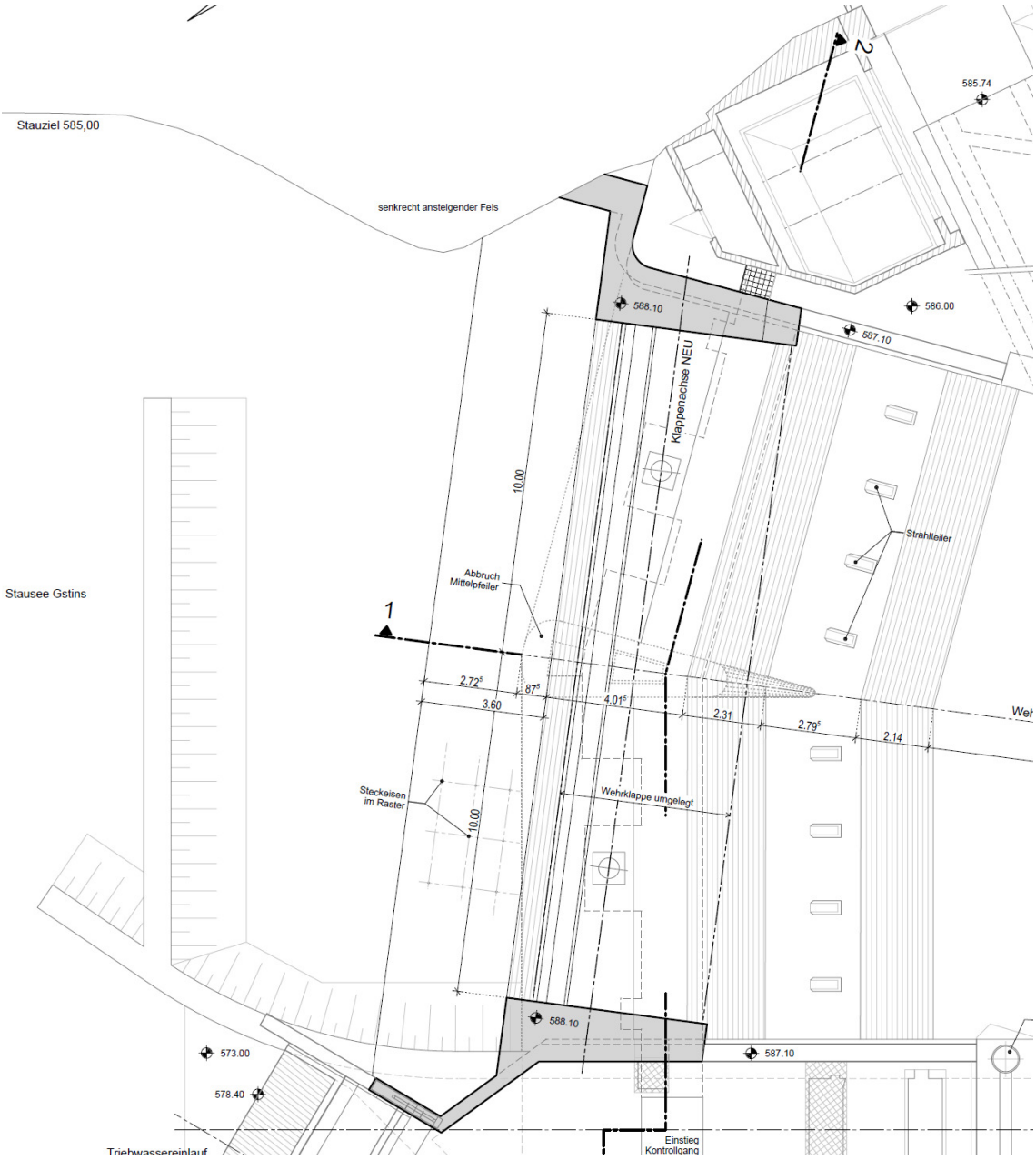


Figure 8.2: *Renovation ground view - Weir - Vorarlberger Kraftwerke AG [2013]*

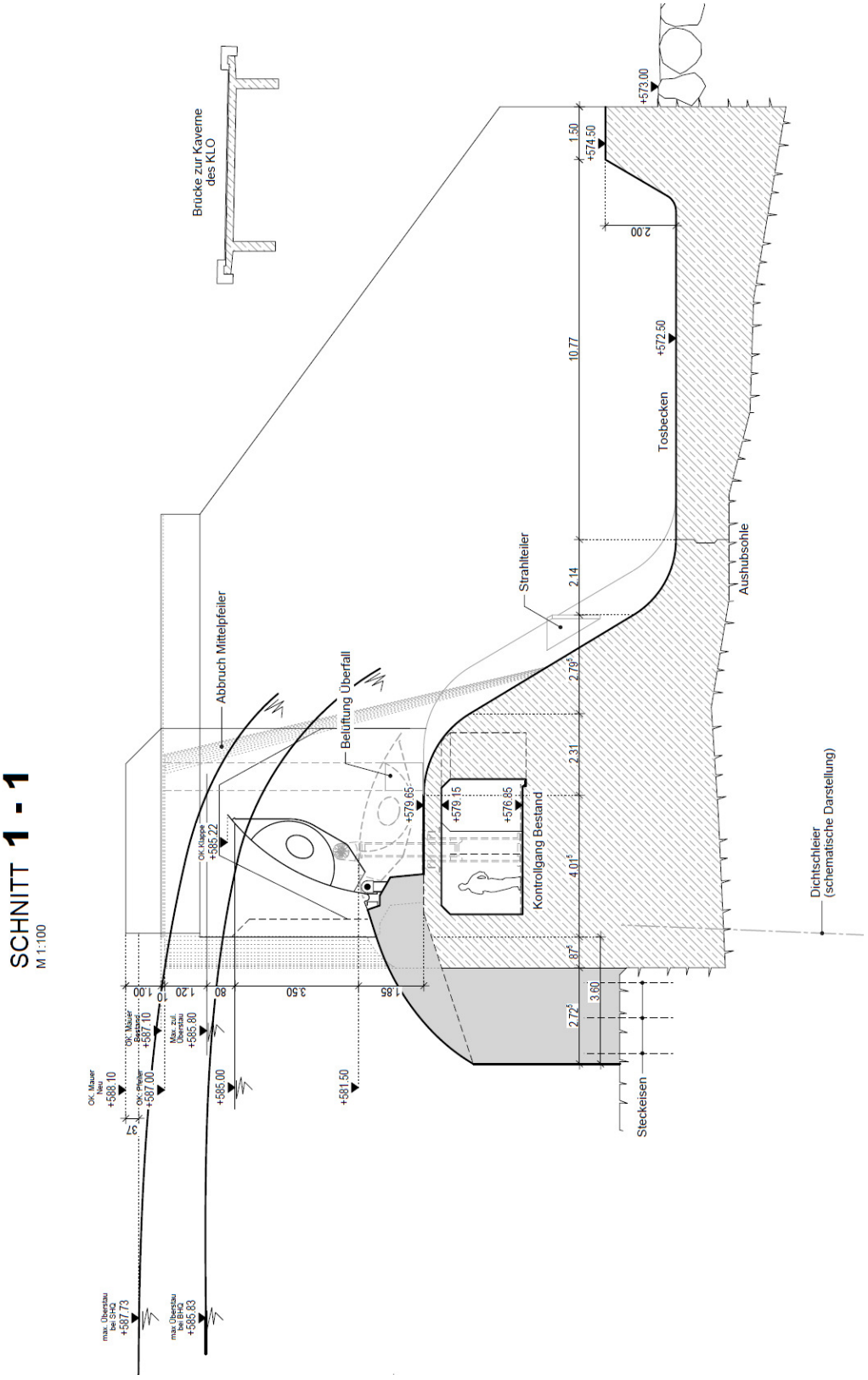


Figure 8.3: *Renovation Cross Section - Vorarlberger Kraftwerke AG [2013]*

8.2.2 Photographs

Figure 8.4: *Existing weir structure - 1 - 19.03.2013 - Vorarlberger Kraftwerke AG [2013]*

Figure 8.5: *Existing weir structure - 2 - 19.03.2013 - Vorarlberger Kraftwerke AG [2013]*

Figure 8.6: *Existing weir structure - 3 - 23.12.2012 - Vorarlberger Kraftwerke AG [2013]*

Figure 8.7: *Existing weir structure - 4 - 23.12.2012 - Vorarlberger Kraftwerke AG [2013]*

8.3 Physical model test

8.3.1 Plans

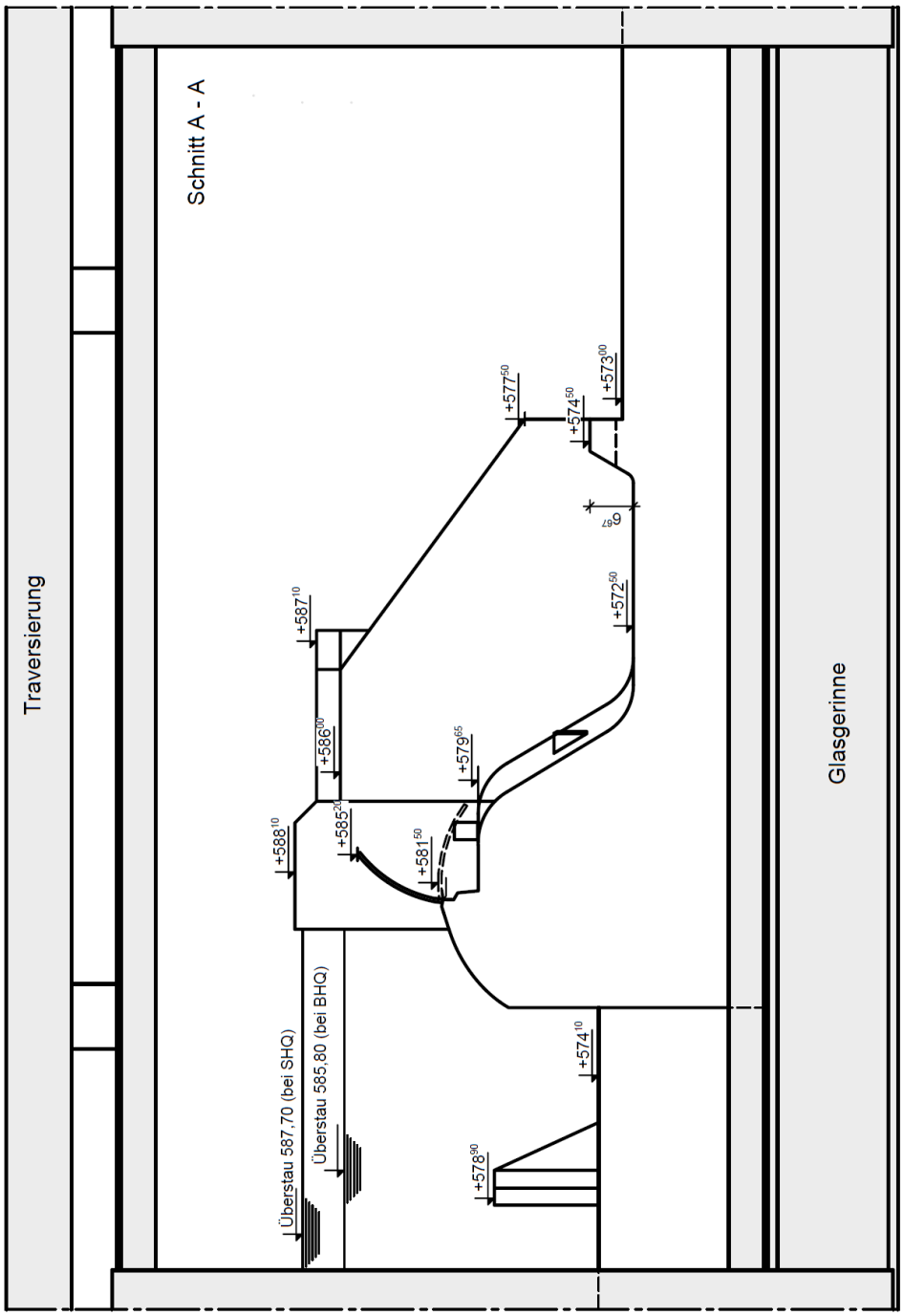


Figure 8.8: *Plan model test - Cross section - IWB [2014]*

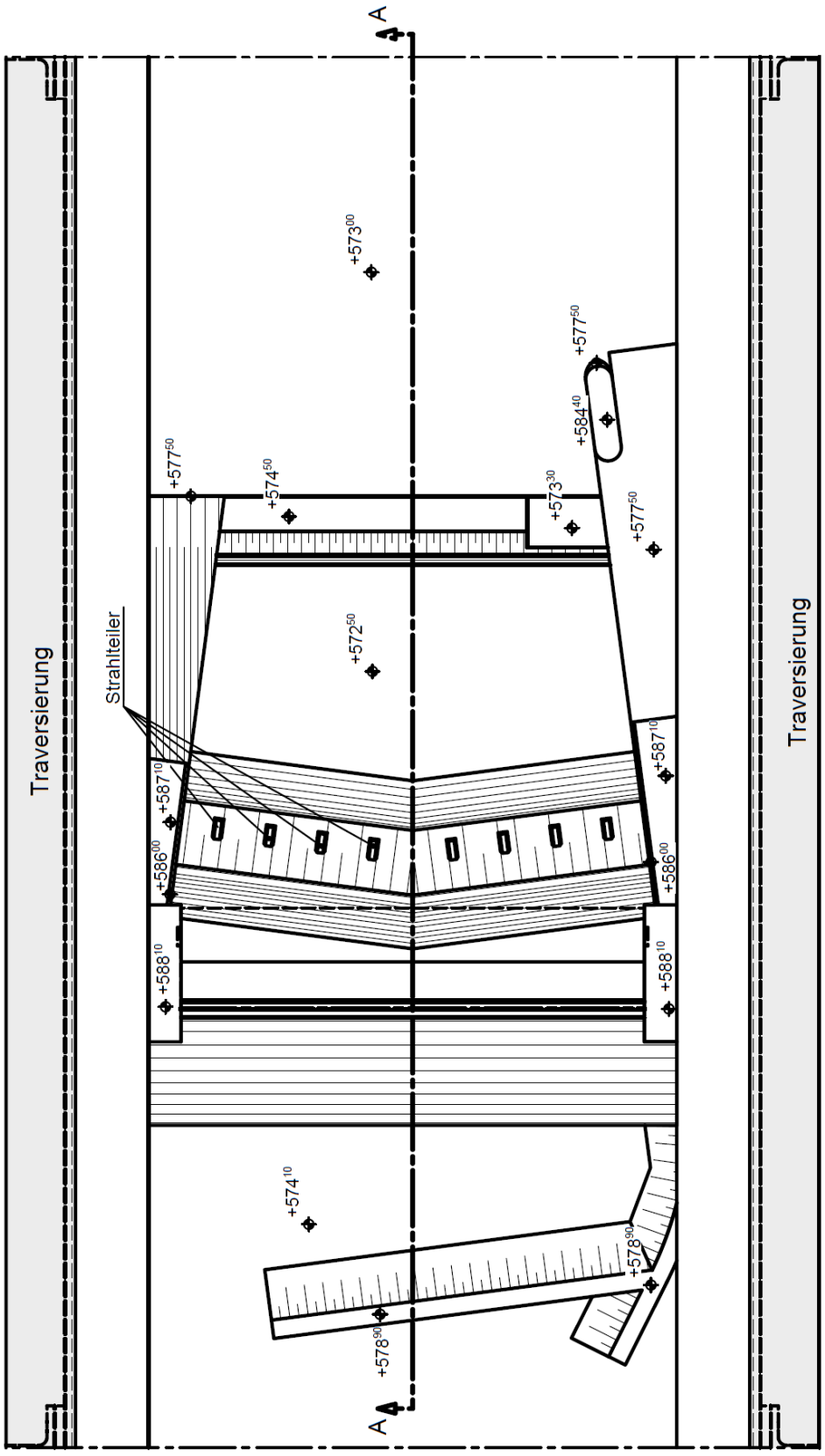


Figure 8.9: *Plan model test - Ground view - IWB [2014]*

8.3.2 Photographs of operated model

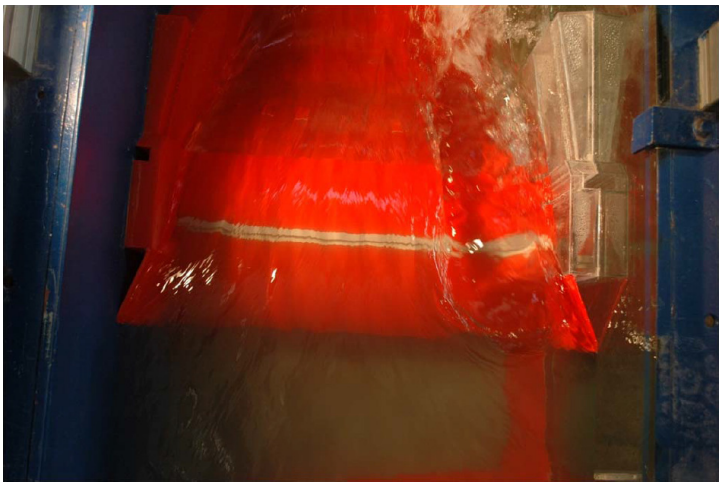


Figure 8.10: *Side pillar version 1 - Top view - IWB [2014]*

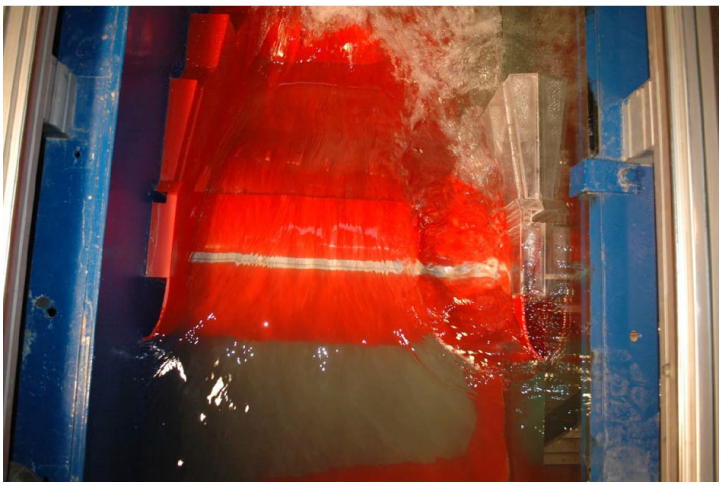


Figure 8.11: *Side pillar version 1 - Top view - IWB [2014]*

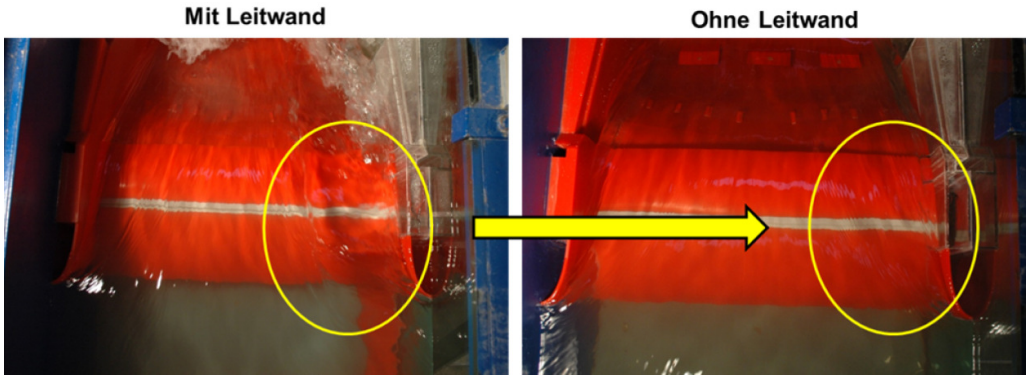


Figure 8.12: *Influence of diversion dam - Top view - IWB [2014]*

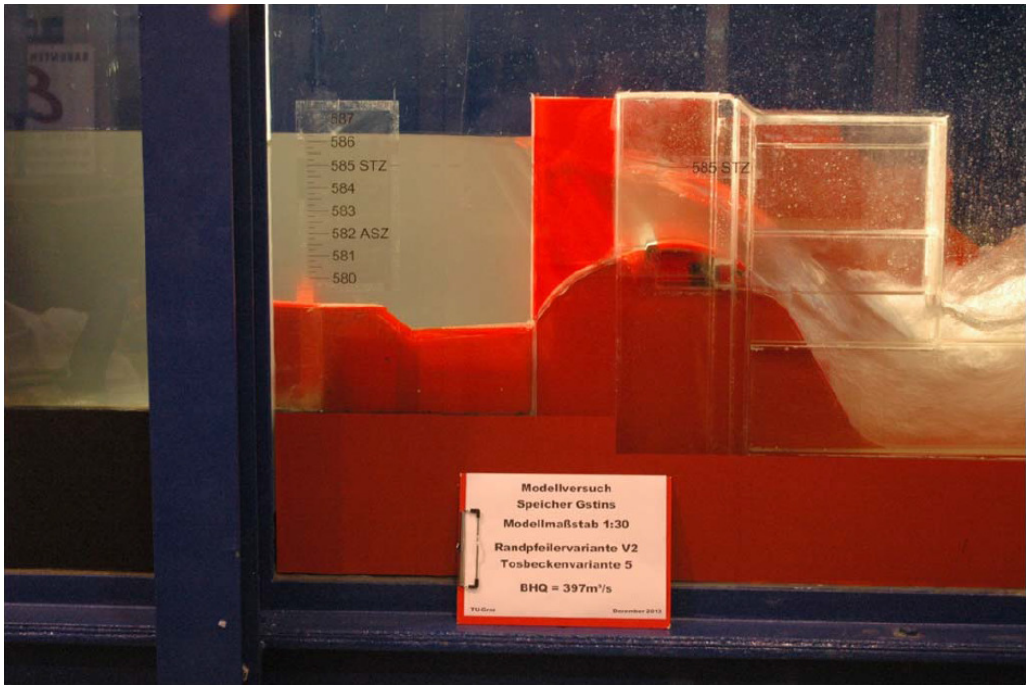


Figure 8.13: *BHQ operation - Side view - IWB [2014]*

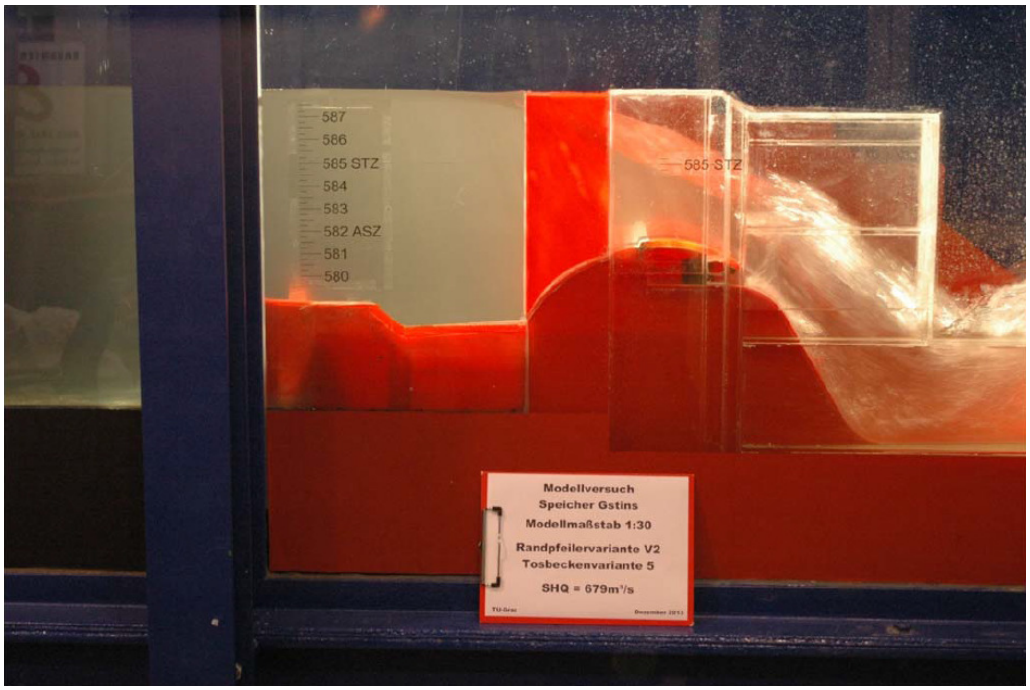


Figure 8.14: *SHQ operation - Side view - IWB [2014]*