



DEGREE PROJECT IN FLUID MECHANICS

SECOND CYCLE, 30 CREDITS

Study of mixing and exchange in a drinking water reservoir using CFD modeling

NADJA RABIZADEH

Authors

Nadja Rabizadeh, nadjara@kth.se
Engineering Mechanics
KTH Royal Institute of Technology

Place for Project

Stockholm, Sweden
KTH Royal Institute of Technology

Examiner

Philipp Schlatter
KTH Royal Institute of Technology

Supervisors

Daniel Hellström
Norrvatten

Lars Jonasson
DHI Sverige

Timofey Mukha
KTH Royal Institute of Technology

Abstract

This thesis examines water mixing and exchange in a drinking water reservoir operated by the municipal association Norrvatten. Recent water samples from the reservoir's outgoing water have shown an increase in culturable bacteria during late summer and fall. This thesis utilizes Computational Fluid Dynamics (CFD) modeling and analysis in OpenFOAM to simulate reservoir inflow and outflow, analyzing mixing processes and their relationship to operational strategies. The objective is to understand the correlation between the residence time of water and microbial growth and propose operational improvements to increase the exchange of water in order to achieve improved water quality. A trace element was implemented in the CFD model to simulate the residence time of water. Initial simulations were based on the reservoir's historical operational data, utilizing temperature and water level measurements provided by Norrvatten. After the initial simulations, four alternative simulations were performed, comparing different operational strategies by modifying inflow parameters. Inflow parameters that were changed were the volumetric inflow rate, water level variation, and the temperature of the inflowing water. The post-processing in ParaView focused on the thermal stratification and residence time distribution near the outlet during each mixing process. The study revealed a complex relationship between flow conditions and microbial growth, making it challenging to identify a clear pattern. However, based on the simulations with the alternative operational strategies it was concluded that the set of operational strategies called "Strategy 1" generated the most optimal flow conditions. This strategy involves a three times larger volumetric inflow rate (an increase from 0.05 to 0.15 m³/s) and a water level that is kept at the same values compared to the original simulation. Strategy 1 resulted in a 3.6 % higher water exchange compared to the original simulation. In comparison to the other simulated strategies, Strategy 1 generates the highest water exchange, with a 63.6 % increase compared to the worst-case scenario involving colder inflow. The conclusion that could be drawn is that the most favorable operational strategies involve higher volumetric inflow rates, lower water levels, and an incoming water temperature which is higher than the initial reservoir temperature.

Sammanfattning

Denna avhandling undersöker omblandningen och utbytet av vatten i en dricksvattenreservoar som drivs av den kommunalförbundet Norrvatten. Nyligen tagna vattenprover från reservoarens utgående vatten har visat en ökning av odlingsbara bakterier under sensommar och höst. Computational Fluid Dynamics (CFD) modellering och analys används i OpenFOAM för att simulera inflöde och utflöde i reservoaren, analysera blandningsprocesser och deras relation till använda driftstrategier. Målet är att undersöka sambandet mellan vattnets uppehållstid och mikrobiell tillväxt samt föreslå operativa förbättringar för att öka vattenutbytet och förbättra vattenkvaliteten. Ett spårelement implementerades i CFD modellen för att simulera vattenets uppehållstid. Inledande simuleringar baserades på reservoarens historiska operativa data och utnyttjade temperatur- och vattennivåmätningar av Norrvatten. Efter de inledande simuleringarna utfördes fyra alternativa simuleringar där olika driftstrategier jämfördes genom att ändra på inflödesparametrar. Inflödesparametrar som ändrades var den volymetriska inflödes hastigheten, variationen i vattennivå och temperaturen på inflödande vatten. Analysen av simuleringarna i ParaView fokuserade på temperaturskiktning och fördelning av vattnets uppehållstid nära utloppet under varje blandningsprocess. Studien avslöjade ett komplex samband mellan flödesförhållanden och mikrobiell tillväxt. Men baserat på simuleringarna med de alternativa driftstrategierna drogs slutsatsen att den uppsättning driftstrategier som kallas "Strategi 1" genererade de mest optimala flödesförhållandena. Denna strategi innefattar en tre gånger större volymetrisk inflödes hastighet (en ökning från 0,05 till 0,15 m³/s) och en vattennivå som varierar på samma sätt som i den ursprungliga simuleringen. Strategi 1 gav ett 3.6 % större vattenutbyte jämfört med den ursprungliga simuleringen. Jämfört med andra simulerade strategier, genererar Strategi 1 det högsta vattenutbytet med en ökning på 63.6 % jämfört med det värsta scenariot som innefattar kallare inflöde. Slutsatsen som kunde dras är att de mest fördelaktiga driftstrategierna är de som innefattar en högre volymetrisk inflödes hastighet, lägre vattennivå och en temperatur på inflödande vatten som är högre än reservoarens initiala temperatur.

Acknowledgements

I would like to give a special thanks to my supervisor Daniel Hellström at Norrvatten for allowing me to work on a project related to my interest in fluid mechanics, while contributing to the research in the socially important field of drinking water distribution systems. Thank you for your guidance during the project. I would also like to thank Linda Holmer at Norrvatten who offered me help with understanding the microbiological part of the project.

I would like to thank my supervisor Lars Jonasson at DHI for introducing me to OpenFOAM and Linux, and for helping me structure the project.

A special thanks to my examiner Philipp Schlatter at KTH for his support and for choosing to review this degree project.

I would also like to extend my deepest gratitude to my academic supervisor Timofey Mukha who has been the most helpful during this thesis project. Thank you for sharing your CFD expertise with me and for giving me advice and feedback when I was facing obstacles. Thank you for your patience, help, and kindness. I could not have asked for a better supervisor.

Lastly, I would like to express my gratitude towards my family and my dear boyfriend. Thank you for always being by my side and for believing in me.

Contents

1	Introduction	1
1.1	Background	2
1.2	Problem formulation	2
1.3	Purpose and goal	3
1.4	Delimitations	3
1.5	Outline	4
2	Theory	6
2.1	Microbiology of drinking water	6
2.1.1	Drinking water regulations	7
2.1.2	Heterotrophic bacteria	7
2.1.3	Disinfection	8
2.1.4	Microbiological water analysis	9
2.1.5	Flow cytometry	10
2.1.6	HNA cell counts as an indicator of microbial activity	13
2.2	Hydraulics of drinking water storage tanks	15
2.2.1	Turbulent flows	15
2.2.2	Thermal stratification	16
2.2.3	Exchange of water	16
2.3	Computational fluid dynamics	17
2.3.1	Finite volume method	17
2.3.2	Navier-Stokes Equations	18
2.3.3	Turbulent flow modeling	18
2.3.4	Boussinesq approximation	22
2.3.5	Heat transfer	24
2.3.6	CFD modeling in OpenFOAM	26

2.3.7	Boundary layer treatment	33
2.4	The reservoir	40
2.4.1	Operational conditions	40
2.4.2	Water sampling	40
2.4.3	Water level	41
2.4.4	Temperature	42
2.4.5	Microbiological measurements of the reservoir water	43
3	Methods	47
3.1	CFD setup specification in OpenFOAM	47
3.1.1	Physical setup	47
3.1.2	The reservoir model	48
3.1.3	Mesh generation process	49
3.1.4	Boundary conditions	52
3.1.5	Wall functions	56
3.1.6	Solver setup	57
3.2	Mesh independence study	58
3.2.1	Temperature averaged over outlet	59
3.2.2	Time-averaged temperature profile	60
3.3	Simulations of operational strategies	61
3.3.1	Simulation case setup and approach	61
3.3.2	Initial conditions	64
3.4	Simulation with alternative operational strategies	67
3.4.1	Simulation case setup	67
3.5	Post-processing	69
3.5.1	Residence time of outgoing water	70
3.5.2	Vertical distributions	70
3.5.3	Comparison with flow cytometer data	71
4	Results	72
4.1	Simulations of operational strategies	72
4.1.1	Temperature distribution	72
4.1.2	Residence time distribution	73
4.1.3	Profiles at outlet	76
4.1.4	Residence time at outlet	77

4.2	Alternative operational strategies	80
4.2.1	Temperature distribution	81
4.2.2	Residence time distribution	81
4.2.3	Residence time at outlet	83
4.2.4	Temperature at outlet	84
5	Discussion	86
5.1	Limitations	86
5.2	Failure analysis	87
5.3	Comparison with flow cytometer data	88
5.4	Exchange of water	89
6	Conclusions	91
6.1	Future work	92
	References	94

Chapter 1

Introduction

One step in the process of distributing drinking water is to store the water that has been transported from the drinking water treatment plant (DWTP) where it has been cleaned. Part of the water is stored in drinking water storage tanks, called reservoirs, before being sent to the distribution system. The main aim of storing drinking water is to even out the variations of water consumption during the day [1]. Another benefit of storing drinking water is to have an emergency reserve in case of disruption in the DWTP or in the case of firefighting where great withdrawals of water are needed. By storing drinking water higher delivery security can be obtained and it helps maintain the required pressure level in the pipeline network. A disadvantage of storing drinking water is that it becomes more exposed to microbial growth because of conditions that may arise that promote the growth of microorganisms. As the water is being stored, thus becoming older, the amount of excess chlorine decreases, leading to increased levels of bacteria in the water. Hence, the drinking water quality may become deteriorated.

To ensure that the stored water in a reservoir is of good quality and meets the requirements for drinking water, the mixing and exchange of water needs to be optimized. The exchange of water is the replacement of old water with new incoming water. The age of the water is defined as its residence time in the reservoir, meaning that water that has just entered the reservoir has no age. As the water resides in the tank, it gets older.

This thesis report covers the investigation and analysis of water mixing and exchange in a reservoir outside Stockholm. The reservoir is a drinking water storage tank run by

the municipal association Norrvatten.

1.1 Background

Norrvatten produces and distributes drinking water to 14 municipalities in north Stockholm and is the fourth largest producer of drinking water in Sweden. Raw water is pumped from Mälaren into the drinking water treatment plant Görvålnverket, which is located by Mälaren in Järfälla [2]. Here, the raw water is cleaned and drinking water is produced. After the cleaning process, the water is pumped out into the distribution system which consists of pipelines and reservoirs.

The reservoir that is to be analyzed in this project is a water tower located above ground. It is divided into two sections, one outer circular geometry which encloses an inner snail-shaped geometry. The two sections of the reservoir have the same volume and can be operated independently of each other. Each section has a separate vertical inlet and outlet located at the bottom of the tank. The work of this project is only applied to the inner section of the reservoir. Henceforth, when mentioning "the reservoir" it will refer to the inner section of the reservoir.

1.2 Problem formulation

In connection with the *Svenskt Vatten Utveckling* (SVU) project called *Rationell mikrobiell analys för biostabilt dricksvatten* [3], it has been observed that outgoing water from the reservoir has demonstrated an increased number of culturable bacteria during late summer and fall (July-October) [4]. During these periods, water samples have been assessed as "potable with reservation" from a microbiological point of view. This means that the drinking water is considered safe for consumption with some microbiological concerns. There is a desire to follow up on these concerns and try to improve the quality of the water in the reservoir.

Factors that are related to the microbial growth in the reservoir are many, one contributing factor is the age of the water. Due to inadequate mixing of the water, the water may become stagnant in some parts of the reservoir volume. Thermal stratification is also a problem that further complicates the mixing of the water, especially when incoming water is colder and thereby has a higher density. The

inadequate mixture of the water leads to an insufficient exchange of water, which causes water to stay longer in the reservoir than desired. When water is stagnant for a longer time, there is an increase in microbial growth.

Insufficient exchange of water can be avoided by looking over the operational conditions of the reservoir, such as the filling and draining of the reservoir. Other parameters that can be controlled to increase the exchange of water are inflow, depth, inlet diameter, and temperature. The control of these parameters and how they affect the exchange of water in the reservoir will be discussed further ahead in the report.

1.3 Purpose and goal

This report covers a master thesis project carried out as the completion of the master's program in Aerospace Engineering at KTH. The project is done for Norrvatten in collaboration with the consultant company DHI and involves the Computation Fluid Dynamics (CFD) modeling of a drinking water reservoir.

The overall purpose of this project is to investigate the relationship between microbial growth in the reservoir and the residence time of the reservoir water. This is done by studying the mixing and exchange of water and thermal stratification in the reservoir for different temperature conditions, including worst-case scenarios when water samples show an increase in the growth of bacteria. The project also includes the investigation of how the operational strategies affect the inflow conditions and the movement of the water in the reservoir. The final goal is to identify operational strategies which are favorable by generating an increased exchange of water and thus lowering the levels of microbial growth.

1.4 Delimitations

The CFD modeling used in this project does not incorporate the simulation of bacterial cell growth in water, thus making it impossible to simulate the behavior and dynamics of microbial populations. Therefore, the analysis of simulated microbial growth and its comparison with microbial data (either from flow cytometer or water samples) can not be considered as a way of validating the simulations. Additionally, it is not feasible to simulate the surface film which naturally forms on the surface of the water.

Without simulating the surface film one can not understand how it behaves and relates to the operational strategies and the microbial growth measured in the sampling tap. Therefore it is hard to get an overall picture of what the distribution of microbial growth in the reservoir looks like and where the most problematic microbial communities reside.

This study does not cover the trial and error process of either adjusting the turbulence model or testing and comparing different turbulence models in CFD. This is an essential step toward identifying the most appropriate model for simulating the fluid dynamics of a specific case accurately. Since the choice of turbulence model and solver had already been evaluated and decided on by DHI before the start of the project, this step could be omitted.

1.5 Outline

The thesis report is structured as following:

Chapter 2 presents the theory related to the microbiology and hydraulics of a drinking water storage tank. Additionally, the theory behind the computational fluid dynamics and how it is applied to the specific flow case of the reservoir is covered. The normal operation of the reservoir is also explained in terms of operational conditions, water sampling, and flow cytometer measurements.

Chapter 3 presents the method for setting up the CFD simulation, describes the mesh independence study, and describes the process of setting up different simulation cases. The post-processing approach used to analyze the mixing and exchange of water in the reservoir for different operational cases and for correlating the results with experimental data from Norrvatten is also explained.

Chapter 4 presents the relevant results of the CFD simulations and highlights the important findings of the results and how they connect to the purpose of the study.

Chapter 5 covers the discussion of the results and the overall work of the project. Here, important correlations between operational strategies and flow conditions are discussed in relation to literature and experimental data provided by Norrvatten. Furthermore, the limitations of the study are discussed.

Chapter 6 covers the conclusions based on the main findings of the results, as well as

recommendations for future work.

Chapter 2

Theory

2.1 Microbiology of drinking water

In Sweden, the whole field of municipal water and wastewater activities in a town is managed by the local municipality. With 290 municipalities as its members, Svenskt Vatten represents the national municipal units for water and wastewater [5].

The Swedish municipal associations of water and wastewater activities annually produce and distribute around 0.9 billion cubic meters of drinking water with quality requirements based on the drinking water regulations of the Swedish Food Agency (Livsmedelsverket) (SLVFS 2001:30) [6]. In today's society, there is a lot of expertise about how to effectively clean raw surface water and groundwater in water treatment plants and also how to distribute the water to consumers with the help of complex hydraulic distribution systems. However, there is a lack of knowledge regarding how the drinking water quality in different parts of the distribution system can be affected, and how to maintain and control a stable water quality throughout the system [4].

The growth of microorganisms in drinking water is something that needs to be controlled and minimized as much as possible to ensure that the water holds a sufficiently good quality. Microorganisms include bacteria, fungi, and viruses. Both biofilm and planktonic bacterial communities reside in a drinking water distribution system, where at least 98 % of the bacterial biomass corresponds to biofilm bacteria. The biofilm is a coating of microorganisms located on solid surfaces that are in contact with water. It is estimated that the biofilm consists of 100 million fixed bacterial cells per square centimeter of pipe surface [7]. Along the water's journey through

the pipeline network, its contact time with the biofilm increases, which results in the biofilm naturally releasing planktonic bacteria. Planktonic bacteria are free-living bacteria that float or swim in the water. Biofilm bacterial communities in distribution systems are resistant to external influences such as disinfection and flushing. The biofilm bacteria mainly consist of oligotrophic bacteria which are well adapted to low temperatures and nutrient-poor water.

2.1.1 Drinking water regulations

The Swedish Food Agency has established regulations to ensure that the drinking water in Sweden is safe and free from harmful bacteria. These regulations include limits for slow-growing bacteria and heterotrophic bacteria in drinking water. The limit values where the drinking water should be assessed as “potable with reservations” are defined as:¹

- 100 CFU/ml for heterotrophic bacteria
- 5000 CFU/ml for slow-growing bacteria

CFU stands for a colony-forming unit.

To ensure compliance with these regulations, public water systems in Sweden are required to regularly test their water for slow-growing and heterotrophic bacteria [6].

2.1.2 Heterotrophic bacteria

The majority of microorganisms in drinking water are heterotrophic, meaning that they are dependent on already existing organic material and thus thrive in organic environments where they can take nutrition from other sources of organic carbon [8]. The conditions of a water storage tank may favor the growth of heterotrophic bacteria since the incoming water that has been transported from the DWTP may have a high residual content of bioavailable organic material, which bacteria feed on. The growth of bacteria in the water is exponential during favorable conditions.

¹It should be clarified that the limit values and assessments reported in this report are based on the previous regulations (which applied in 2022 when the main part of the work was carried out). The Swedish Food Agency’s new regulations on drinking water entered into force on 1st January 2023.

Heterotrophic plate counts

Culturable heterotrophic bacteria have become a common indicator for assessing the general quality of distributed water. The heterotrophic plate count (HPC) method is used to measure the population of heterotrophic microorganisms by counting the colonies in a sample. The number of colony-forming units is counted per milliliter of water. A colony-forming unit estimates the number of viable microbial cells in a sample that can reproduce. With the HPC method, the number of cells in a sample is counted based on their ability to produce colonies under certain conditions of temperature, nutrient medium, and time (defined incubation conditions). There are some uncertainties with expressing results as CFUs since it cannot be said if a colony has been produced by one cell or a group of cells [9]. According to Staley and Konopka [10], the HPC method only enumerates a small fraction of the total direct microscopic count of bacteria in aquatic environments. Studies have shown a great deviation between the viable plate count and the total direct microscopic count, which may have several explanations. In addition to that non-viable cells may exist in the community, there also might exist viable cells that are not reproductive and therefore not counted in the heterotrophic plate count. Furthermore, studies showed that obligate oligotrophs, which are an important part of the planktonic bacterial community, grow poorly in a laboratory and are difficult to study using conventional laboratory equipment and procedures.

2.1.3 Disinfection

The purpose of disinfection is to ensure that no pathogenic bacteria exist in the treated water and also to limit any uncontrolled microbial regrowth during distribution. At the DWTP Görälnverket, the drinking water is primarily disinfected with UV disinfection which helps kill water-borne pathogenic microorganisms. Other hygienic barriers that can be used before distribution are ozonation, membrane filtration, and chlorine-containing disinfectants. During treatment and distribution, it is common for bacteria to regrow. Even though most of the bacteria reside and proliferate in biofilms, it is still normal that drinking water contains concentrations of 10^4 - 10^5 cells/ml of diverse microbial populations [11]. Chloramine can be used as a residual disinfectant to keep down the regrowth of bacteria since it does not give as harmful by-products as chlorine and also does not break down as easily [4]. There is a relation between low excess of

chlorine in the water and increased levels of bacteria. When the water is getting older, the excess chlorine will decrease because of its reactions with organic and non-organic substances in the water. The increase in the water's age will thus favor microbial growth.

It is important to highlight that some by-products from disinfection may be harmful to the water quality. According to the regulation SLVFS 2001:30, § 3 [6], distributors of water are required to control that the disinfection is effective and that any pollution arising from by-products are minimized as much as possible without risking the efficiency of the disinfection. Unwanted disinfection by-products can be organochlorine compounds, bromate, ammonium, and nitrite [1]. When chlorinating drinking water containing organic material, potentially harmful substances can be formed such as organochlorine compounds (due to the increased decomposition of organic material). This may affect the biostability of the water. Studies in the *Svenskt Vatten Utveckling* (SVU) project have shown that the gradual elimination of monochloramine as secondary disinfection resulted in a decreased amount of unwanted biofilm bacteria in the distribution system [4]. In the project, 16 distribution points part of the distribution system in Varberg were analyzed.

2.1.4 Microbiological water analysis

One can assess the drinking water quality by examining the microbial activity in the water. The traditional analytic method HPC, described in section 2.1.2, is a common approach. Water samples are collected and analyzed in a laboratory under controlled conditions, to either identify different microorganisms (qualitative method) or to estimate the number of microorganisms present (quantitative method). Since it is not possible to examine all occurring microorganisms in a water sample, some types of organisms, called indicator organisms, are examined to measure the microbial activity in a water sample. The indicator organisms are types of bacteria that can be used to detect and estimate the level of contamination of the water, by indicating whether pathogens are present or not. Indicator organisms are easy to cultivate and identify and have a better chance of survival. Even though they are non-pathogens, they may still be present in the same environments as pathogens [8]. The qualitative method is mainly used to see if pathogenic microorganisms are present, such as the bacterium *E. coli*. The quantitative method involves a bacteriological water analysis which is a

method to estimate the number of bacteria present.

In the laboratory, the culturable bacteria in a water sample will be exposed to an environment that encourages cell growth, in order to grow microbial cultures. The microbial cultures are cultivated according to standard procedures, either for three or seven days. When the cultivation is done, the number of colonies grown in the sample is counted, indicating the microbial activity in the water sample. The microbiological water analysis is statutory and is therefore done regularly by distributors of drinking water.

3-days microbial cultivation

The number of heterotrophic bacteria colonies is counted after 3 days of incubation at 22 degrees ° C. The results from the cultivation show different types of bacteria that can grow in drinking water. It also shows, although in comparatively small quantities, fungi and actinomycetes [8].

7-days microbial cultivation

The methodology for the 7-day microbial cultivation is the same as for the 3-day, the only difference is that the incubation time is extended to 7 days. This is done to study the amount of slow-growing bacteria in the water. Slow-growing bacteria may include various types of bacteria, and they are used to indicate the growth potential for bacteria in water reservoirs and in the pipeline network [8].

2.1.5 Flow cytometry

Flow cytometry (FCM) is a rapid and accurate technique used to detect and measure the physical and chemical properties of bacterial cells in drinking water, such as the cell size and cellular properties needed to characterize bacteria. FCM is based on laser detection of stained bacteria in water samples. The bacterial cells are stained with fluorescence-staining dyes that bind to the DNA in the cells. Bacterial cells suspended in a hydrodynamically focused liquid stream pass through a pulsed beam of laser light [12]. Scattered laser light and fluorescent emissions are collected by optical detectors. The signals produced in the FCM are then digitized for further computational analysis of light scatter data and fluorescence data. The light scatter data provides information about the physical properties of a cell, such as size, shape, and surface features. The

fluorescence data reveals the autofluorescence of the cells which helps characterize bacteria, distinguishing it from background noise and demonstrating cell viability and vitality.

Detection of bacterial cells

Gating is a crucial step in the analysis of flow cytometry data for bacterial cells. It involves the selection of specific cell populations from a heterogeneous sample based on their physical and/or biological properties. Gating can be used to identify different subpopulations based on parameters such as size, granularity, and fluorescence intensity. For example, one can use forward scatter (FSC) and side scatter (SSC) to identify bacterial cells based on their size and complexity, respectively. Fluorescently labeled antibodies or dyes can also be used to identify specific bacterial subpopulations based on their fluorescence intensity [12].

One common approach to discriminate bacteria from debris particles and background signals in a water sample is to plot forward- or side-scatter data against the fluorescence signal of the stain [12]. Another alternative for discrimination of bacteria is to dye the bacteria with a combination of two fluorescence signals (for example red and green fluorescence). This is a favorable approach for natural aquatic environments (that usually contain bacterial cells with varying content of nucleic acid) since it enables visualizing and differentiating between bacteria with high and low nucleic acid content (HNA and LNA). Nucleic acids are large biomolecules that are essential in cells and viruses and their function involves the storage and expression of genomic information [13]. By being able to differentiate between bacteria with high and low nucleic acid one can characterize different types of bacteria in a water sample.

Flow cytometry at Norrvatten

Norrvatten uses two different types of flow cytometers, the laboratory instrument BD Accuri C6 Plus (BD Biosciences, Belgium) and the automatic online instrument Bactosense (Sigrist, Switzerland). BactoSense was acquired year 2020 to automatically analyze water samples in order to discover deviating levels of bacteria, instead of doing it manually in the laboratory. The Bactosense FCM can be placed at different points of a water distribution system [14].

Two different fluorescence staining dyes are used when staining bacteria, SYBR®

Green I and Propidium iodide (PI). They bind to the DNA of the cells and emit light in specific wavelengths, SYBR® Green I emits green light and propidium iodide emits red light. SYBR® Green I stains all bacterial cells and does not differentiate between intact and damaged cells, which is why PI is added. Due to its larger molecular size, PI does not penetrate cells with intact cellular membranes, meaning that it only stains damaged bacterial cells that have lost their membrane integrity. Staining cells with PI is useful when monitoring cell viability after an oxidative disinfection process, as it indicates severe membrane damage and therefore works as an indicator of cell death. The staining method described above makes it possible to visually distinguish bacteria from background particles and debris, and intact bacteria from damaged bacteria [4][12].

The software FlowJo is used to analyze the data and characterize different types of bacteria. The intensity of the fluorescent staining of bacteria reveals information about the amount of DNA in the bacterial cells, making it possible to divide them into bacteria with high DNA content (high nucleic acid, HNA) and bacteria with low DNA content (low nucleic acid, LNA). By visually differentiating between bacteria with a high and low nucleic acid content, one can characterize different bacteria communities. Most importantly, the FCM approach works as a bioindicator of water quality evaluation. A large quantity of HNA bacteria can indicate poor water quality. When analyzing the data gathered from the BactoSense flow cytometer at Norrvatten, one important parameter is HNAP, which reveals the percentage of HNA bacteria. BactoSense also measures the total cell count (TCC) and intact cell count (ICC) [4].

The BactoSense flow cytometer

The BactoSense, developed by SIGRIST [15], is a highly effective flow cytometer. Automatic, continuous online monitoring of TCC in water can be done in 30 min (or longer) intervals. The measurement and analysis process of the BactoSense can be described below in the following steps [16].

1. **Staining:** A sample of water is stained with a fluorescent dye, usually SYBR Green I, which will bind to the DNA of the bacterial cells.
2. **Mixing:** The dye is distributed evenly throughout the sample to guarantee reproducible results.
3. **Incubation:** The sample is incubated and heated to 37 °C for 10 minutes, this

enables the dye to bind to the DNA of each microbial cell.

4. **Measurement:** The sample is measured.
5. **Detection:** One laser diode, at 488 nm, is used to detect the dyed microbial cells. Three detectors measure the fluorescence simultaneously, at a wavelength of 525 nm for FL1, at 715 nm for FL2, and 488 nm for the Side Scatter (SCC). The detection limit is 100 cells/mL- 5000000 cells/mL.
6. **Analysis and reporting:** There is a user interface, in the form of a touch screen, which enables the user to change settings such as the measuring interval. Measured data is permanently stored in the internal database and can be retrieved and visualized to show the characteristic fingerprint of the sample analyzed.

2.1.6 HNA cell counts as an indicator of microbial activity

When enumerating bacterial cells of samples taken in aquatic systems by flow cytometry, it is possible to sort the total counted cells into subgroups of HNA cell counts and LNA cell counts. Generally, LNA cells can be described as inactive and nonviable cells with damaged membranes; the DNA of these cells is degraded [17]. HNA cells are described as intact cells, containing at least a single genome. By staining samples with the fluorescent dye, HNA cells can be discriminated from LNA cells. Dyed HNA cells have an increased fluorescent intensity compared to LNA cells.

The HNA subgroup includes cells with a wide range of activities and may consist of inactive and rapidly growing cells [17]. Distinguishing the most active cells from less active cells and identifying the diversity of species within a subgroup is a challenge. Nevertheless, it is important to be able to identify the contributions to the bulk activity of the whole bacterial community to understand the correlation between the specific microbial activity of a subpopulation and the total bacterial production. This can be done by investigating relationships among cell size, activity, and genetic diversity.

Research involving cell sorting experiments done in seven different aquatic environments showed that HNA cells are responsible for a large portion of the bulk activity [17]. The seven samples (which were collected from freshwater, brackish water, and marine water ecosystems) were stained with the fluorescent dye SYBR II. By leucine incorporation, bacteria were labeled with the amino acid ^3H -leucine in order

to estimate the specific activity of each cellular category. Then HNA, LNA, and total cells were sorted by flow cytometry. Results of the cell sorting showed that the specific activity of HNA cells was higher than the ones of the total and LNA cells. The specific activity of LNA cells was generally very low in comparison with the one of HNA cells. This confirms the hypothesis which states that LNA cells include an important portion of dead cells due to most of them not responding to environmental changes. The average contribution of HNA cells to the total activity of the community was sometimes close to 100 percent.

Results from the seven samples differed due to the different environments. The less active bacterial communities had a somewhat higher contribution of LNA cell activity to the total production. For these samples, the differences between the specific activities of the HNA and LNA cells were the lowest. This was likely due to poor discrimination between the two subgroups, which had been observed in the samples with less active communities. Similar poor discrimination between the two subgroups has also been reported by Gasol and del Giorgio [18], which may be because the subgroups are not properly separated. To trust the results of the HNA cell counts, the objective separation of the subgroups should be improved.

In Fig. 3, found in the report written by Lebaron et al., [17], it can be seen that the correlation between the HNA cell activity and the total bacterial production from the seven samples is very good. The r -squared value r^2 is close to one, meaning that there is almost a perfect fit of the sample data to the regression line. All seven points, each corresponding to data from a sample, add up to a perfect positive correlation. The ratio of change is constant which means that the rate of change of the microbial activity of HNA cells corresponds to the rate of change of the total microbial production of the whole bacterial community. On the other hand, for another group of samples, corresponding data showed that the correlation was not as good [17]. For this case, the r -squared value r^2 was equal to 0.36, which classifies (on the lower bound) as a modest correlation. Based on these results, it can be concluded that the microbial activity within the HNA cell subgroup is heterogeneous. Not all of the cells are active or have high cell activity. An explanation for this may be the great variation of cell volumes within the HNA cell subgroup.

A study of relationships between cell size and productivity in natural aquatic communities showed that cells within the medium cell-size class ($0.31\text{--}0.36\ \mu\text{m}^3$) are

the fastest growing [19]. The study involved the flow sorting of microorganisms from different environmental samples. A bell-shaped relationship was found between cell biovolume and activity, which can be seen in Fig. 3 c) in the report written by Bernard et al. [19]. From this, it can be concluded that small cells are generally inactive cells, and also that a diversity in cell size of a population will also give a diversity in cell activity. An increase in cell size may increase bacterial growth, but not in all cases. Based on the two studies discussed above, it was concluded that HNA cells may be used as an indicator of microbial activity in aquatic environments.

2.2 Hydraulics of drinking water storage tanks

The fluid motion of water inside a storage tank depends strongly on the inner geometry and the flow characteristics in relation to the governing equations. Fluid flows are governed by the Navier-Stokes equations which are non-linear second-order partial differential equations (PDEs). The equations are based on the physical conservation laws (conservation of mass, momentum, and energy). The principle of conservation states that certain physical measurable quantities are conserved over the whole computational domain. By discretizing the Navier-Stokes equations using the Finite Volume Method (FVM), the equations can be converted into a system of algebraic equations which can be solved with computational fluid dynamics (CFD).

2.2.1 Turbulent flows

Turbulent flow is a complex and dynamic phenomenon that occurs in many different fluid systems, including drinking water reservoirs. Turbulent flow can have a significant impact on the quality of the water that is supplied to consumers. Turbulence can create mixing and diffusion of chemicals and microorganisms in the water, which can affect its taste, odor, and overall quality.

Turbulent inflow can occur in drinking water reservoirs when water is introduced into the reservoir through a high-energy jet, such as an inlet structure. The jet stream needs to be turbulent in order to be an effective mixer. At a Reynolds number over 10000, the turbulence is fully developed [20]. The most important characteristic of a turbulent jet stream is its momentum flux. A jet stream that is allowed to develop freely (without being disturbed by walls or obstacles) will have a constant momentum flux in the flow

direction. However, the volume flow rate is not constant but increases with distance from the inlet pipe due to vortices in the jet stream dragging surrounding water. Due to the constant momentum flux, experimental studies have shown that the center velocity of the stream decreases linearly while the width increases linearly with the distance from the inlet pipe discharge. The flow rate in a free jet stream increases by an amount equivalent to the inlet flow rate after every distance of about 3.5 inlet diameters. The increase in flow indicates that inflowing fresh water is diluted with surrounding water to the same degree. The rapid dilution due to the increase in flow rate explains why turbulent jet streams are such effective mixers [21].

The extent and intensity of the turbulence in the reservoir depend on several factors, including the size and shape of the inlet structure, the flow rate of the water, and the depth of the reservoir. The shape of the inlet structure can also have an impact, with more complex designs leading to increased turbulence.

2.2.2 Thermal stratification

Thermal stratification refers to the separation of water into distinct layers of different temperatures, with the colder and denser water at the bottom and the warmer and less dense water at the top. This phenomenon can occur in large bodies of water, including drinking water reservoirs, when the sun heats the surface water while the deeper water remains cold.

One of the main issues with thermal stratification is that it can impact the exchange of water between different layers. When inflowing water is colder than the water in the upper layers, it may not mix well with the warmer water. This can lead to a situation where the colder water remains at the bottom and does not contribute to the overall water flow. This can have negative consequences for the water quality.

2.2.3 Exchange of water

The exchange of water in storage tanks can theoretically be enabled in two ways. One approach is the *Plug flow principle*, which is when water is forced to flow like a plug in parallel paths from inlet to outlet. In an ideal plug flow, the fluid properties are assumed to be constant across any cross-section of the path perpendicular to the axis of the path. The velocity profile of the flow is uniform, meaning that the fluid particles along the vertical cross-section all have the same velocity. This model results in the

assumption that there is no friction at the inner walls of the paths, meaning there are no boundary layers adjacent to the inner walls [22]. Since the flow is viscous, the assumption of no boundary layers is not reasonable, which means that ideal plug flow will never be achieved. But it can be sought to have a fair plug flow in storage tanks, called a near plug flow.

The other way of enabling the exchange of water is through the mixing principle. Incoming water is then mixed with the stored volume in the tank, and old water is successively replaced with new water. The problem here is to ensure that there is enough inflow momentum to avoid stagnant zones in the tank [23].

Due to the snail-shaped geometry of the reservoir, this project will aim to simulate a near plug flow in the reservoir, and the exchange of water will be dependent on the plug flow principle.

2.3 Computational fluid dynamics

In this section, the theory behind CFD will be explained more in detail. The focus will be on explaining the theory that is needed to model the fluid flow in the reservoir, which is the main task of this project. Governing equations required for setting up the CFD simulation according to the physical conditions of the reservoir and case will be derived and explained.

2.3.1 Finite volume method

To express the governing equations behind CFD, the second-order accurate Finite Volume Method is used as a discretization method. FVM is used to discretize the terms in the Navier-Stokes and other scalar transport equations. FVM derives a discrete system of the continuous physical field of fluid flow by approximating the system of partial differential equations (PDEs) that describe the conservation laws in fluid motion.

Consider a general flow field represented by streamlines. The fluid is moving through a finite control volume V fixed in space. The closed surface which bounds the volume is called the control surface S . The fundamental physical principles of the conservation laws are applied to the fluid inside the control volume and to the fluid that is flowing through the control surface. The PDEs are integrated over the control volume to attain

the conservation equations for mass and momentum which initially will be given in integral form as local balance equations. The set of balance equations is discretized with respect to a set of discrete unknowns which will result in a system of discrete equations [24]. FVM relies on the fluxes at the boundaries of the control volumes being conservative (the flux entering a control volume from its neighbor must be the opposite of the flux entering the neighbor from the control volume) and consistent (the discrete flux attained from an interpolation tends to the continuous flux as the mesh size vanishes).

By using the Finite Volume Method in CFD, a geometry domain is discretized in several non-overlapping finite volumes, also called cells. The cells will form a computational grid with unknown flow variables evaluated at the centroids of the cells. The aim of discretizing the PDEs is to convert the non-linear Navier-Stokes equations to a linear matrix form which can be solved with CFD.

2.3.2 Navier-Stokes Equations

The continuity equation derived from the conservation of mass and the momentum equation derived from the conservation of momentum form a system of equations, called the Navier-Stokes equations.

$$\begin{cases} \rho \mathbf{g} - \nabla p + \mu \nabla^2 \mathbf{v} = \rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) \\ \nabla \cdot \mathbf{v} = 0 \end{cases} \quad (2.1)$$

The equations in (2.1) are the Navier-Stokes equations in vector form for an incompressible and viscous fluid. The second-order nonlinear partial differential equations describe the three-dimensional motion of a fluid and provide an approximation of reality. The Navier-Stokes equations could further be expressed as a set of equations, where each equation defines the x-, y-, and z-components respectively.

2.3.3 Turbulent flow modeling

Turbulent flow which is chaotic and time-dependent, has three-dimensional vorticity fluctuations, including a broad range of time and length scales. An alternative to solving the Navier-Stokes equations directly, which is very computationally expensive, is to use statistical averaging techniques.

RANS modeling

Since the turbulent flow is random, the Navier-Stokes equations can be statistically averaged either by spatial, time, or ensemble averaging. The Reynolds-averaged Navier-Stokes (RANS) equations are time-averaged Navier-Stokes equations.

The Reynolds Averaged Navier-Stokes equation can be expressed as

$$\rho \bar{v}_j \frac{\partial \bar{v}_i}{\partial x_j} = \rho g_i - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \frac{\partial \bar{v}_i}{\partial x_j} - \overline{\rho v'_i v'_j} \right), \quad (2.2)$$

where the term $R_{ij} = -\overline{\rho v'_i v'_j}$ (the averaged product of the fluctuating velocities) is called the Reynolds stress tensor. This is an additional stress term that appears due to interactions between various turbulent fluctuations [25]. The Reynolds stress tensor R_{ij} is symmetric and can be separated into diagonal and off-diagonal components.

The Reynolds stresses are unknown, which introduces the closure problem which says that the number of unknowns is larger than the number of equations. There are four equations: the mean continuity equation and three components of the Reynolds Averaged Navier-Stokes equations (Eq. 2.2). And then there are ten unknowns: three mean velocity components ($\bar{\mathbf{v}}$), the mean pressure \bar{p} and six turbulent stresses ($\overline{v'_i v'_j}$). To close the system of mean equations (Eq. 2.2), the Reynolds stresses need to be expressed as a function of the mean flow properties. By introducing a turbulence model the Reynolds stresses can be modelled, which enables the solving of the RANS equations to compute turbulent flow.

Unsteady RANS modeling

For flows that show transient behaviour due to time-dependent boundary conditions, the unsteady RANS (URANS) method is used. If the underlying physics of the flow is of transient behaviour on a larger time scale than the turbulent time scales, then the RANS method can not be used to approach a steady state. Unlike the RANS equations, the URANS equations retain the time derivative of the velocity in the Reynolds-Averaged momentum equation during calculation. This is because the URANS method is based on ensemble averaging, rather than time averaging of the decomposed flow quantities, so there is still a time dependence.

The ensemble average is defined as the average of a series of measurements of N

identical experiments and is given by

$$\bar{\Phi}(x, t) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \Phi_n(x, t), \quad (2.3)$$

where $\bar{\Phi}(x, t)$ is the averaged value.

URANS is developed by separating the time scale of mean motion and the time scale of turbulent motion. When solving unsteady simulations with URANS, one needs to choose a sufficiently small time step Δt which can resolve the transient behaviour of the mean flow. This time step will be much larger than the largest turbulent time scale ($\Delta t \gg T_t$, T_t being the turbulent time scale). URANS makes it possible to capture the large fluctuations of the mean flow. The URANS equations are defined as

$$\rho \frac{\partial \bar{v}_i}{\partial t} + \rho \bar{v}_j \frac{\partial \bar{v}_i}{\partial x_j} = \rho g_i - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \frac{\partial \bar{v}_i}{\partial x_j} - \rho \overline{v'_i v'_j} \right). \quad (2.4)$$

Boussinesq hypothesis

The Boussinesq hypothesis is used as a foundation for eddy viscosity turbulence models and it is used to express the Reynolds stresses in a simpler form. It states that the turbulent Reynolds stresses are related to the mean velocity gradients (mean strain-rate tensor $\overline{S_{ij}}$) via the turbulent eddy viscosity μ_T . The hypothesis assumes that the Reynolds stresses (defined by the diffusion term in Eq. (2.2)) can be modelled by a viscosity term which includes a turbulent viscosity coefficient ($\mu + \mu_T$). With the Boussinesq assumption, the Reynolds stress tensor can be expressed as

$$R_{ij} = -\overline{\rho v'_i v'_j} = -\frac{2}{3} \rho k \delta_{ij} + \mu_T \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial \bar{v}_j}{\partial x_i} \right), \quad (2.5)$$

where δ_{ij} is the Kronecker delta function and k is the turbulent kinetic energy defined in Eq. (2.6) as the total kinetic energy per unit mass of three fluctuating velocity components.

$$k = \frac{1}{2} (\overline{u^2} + \overline{v^2} + \overline{w^2}) = \frac{1}{2} \overline{v'_i v'_i} \quad (2.6)$$

In Eq. (2.6) the velocity vector is denoted as $\mathbf{v} = (u, v, w)$.

The first term in Eq. (2.5) represents the isotropic part of the Reynolds stress tensor. The second term represents the shear components of the tensor and is called the

anisotropy tensor which is defined as

$$b_{ij} = \overline{\rho v'_i v'_j} - \frac{2}{3} \rho k \delta_{ij}. \quad (2.7)$$

It is by definition a symmetric and traceless tensor [26].

The turbulent eddy viscosity μ_T in Eq. (2.5) needs to be calculated to close the equations, and this is done by introducing an appropriate turbulence model.

Turbulence models

There are several classes of turbulence models for solving the RANS equations, all have different levels of approximations involved. The most common RANS turbulence models are classified based on their number of transport equations. The two-equation models have two additional transport equations that need to be solved along the RANS equations. The equations describe the transport of two independent turbulent scalars. For example, the $k - \epsilon$ model uses transport equations for the turbulent kinetic energy k and the dissipation of turbulent kinetic energy ϵ .

The $k - \epsilon$ model uses the following transport equation for the turbulent kinetic energy k :

$$\rho \frac{\partial k}{\partial t} + \rho \mathbf{v} \cdot \nabla k = \nabla \cdot \left[\left(\mu + \frac{\mu_T}{\sigma_k} \right) \nabla k \right] + P_k + P_b - \rho \epsilon + S_k, \quad (2.8)$$

where the first term on the left-hand side represents the time derivative, the second one represents the convection and the first term on the right-hand side represents the diffusion of turbulent kinetic energy [27]. P_k is the production due to mean velocity shear, P_b is the production due to buoyancy, $\rho \epsilon$ is the dissipation rate and S_k is a user-defined source term.

The transport equation for the dissipation of turbulent kinetic energy ϵ is defined as

$$\rho \frac{\partial \epsilon}{\partial t} + \rho \mathbf{v} \cdot \nabla \epsilon = \nabla \cdot \left[\left(\mu + \frac{\mu_T}{\sigma_\epsilon} \right) \nabla \epsilon \right] + C_1 \frac{\epsilon}{k} (P_k + C_3 P_b) - C_2 \rho \frac{\epsilon^2}{k} + S_\epsilon, \quad (2.9)$$

where the first three terms, similar to the previous equation, represent the time derivative, convection, and diffusion of the dissipation of turbulent kinetic energy. The other terms are source and sink terms, which were defined for the previous equation. C_1 , C_2 , and C_3 are empirical model coefficients that vary depending on the variant of the $k - \epsilon$ model (standard, RNG, or realizable). The coefficients are defined according

to the Jones and Launder (1972) paper [28] and may vary for each model because of the different updates of the coefficients. However, modern CFD codes use the most recently updated coefficients since they have been proven to give better performance. These coefficients are defined in Table 2.3.1 below.

Table 2.3.1: Values of the empirical constants of the $k - \epsilon$ turbulence model.

C_μ	C_2	C_2	σ_k	σ_ϵ
0.09	1.55	2.0	1.0	1.3

The turbulent viscosity is modeled as

$$\mu_T = C_\mu \frac{\rho k^2}{\epsilon}, \quad (2.10)$$

and can be solved after the transport equations have been solved for k and ϵ .

2.3.4 Boussinesq approximation

Heat transfer exists in a flow where the fluid varies in temperature from one place to another, the flow is thus nonisothermal. For nonisothermal flows, the fluid properties are normally functions of temperature [29]. For a buoyancy-driven flow with small density changes, the Boussinesq approximation can be used to simplify the Navier-Stokes equations and make the CFD simulation more stable. The Boussinesq approximation states that the density changes can be neglected (density treated as a constant) in the unsteady and convection terms, but in the buoyant force term, the density is treated as a variable. This approximation can be done because density changes in other terms than the buoyant one are much smaller, for a low Mach number flow.

The density can be modeled as

$$\rho = \rho_0 + \nabla \rho, \quad (2.11)$$

where ρ_0 is the reference density and $\nabla \rho$ is the fluctuating density. The Boussinesq approximation is valid when $\nabla \rho \ll \rho_0$. The expression for density in Eq. (2.11) is inserted in the continuity and Navier-Stokes equations in Eq. (2.1), which gives:

$$\nabla \cdot \mathbf{v} = 0 \quad (2.12)$$

$$\rho_0 \left(\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} \times \mathbf{v}) \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{g} \quad (2.13)$$

In Eq. (2.13), the Boussinesq approximation makes an approximation of the density in the unsteady and convection term on the left-hand side, so that $\rho \approx \rho_0$. In the buoyant force term, the density is modeled as in Eq. (2.11), with a constant reference value and a fluctuating part.

The Navier-Stokes equation can be rewritten as

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} \times \mathbf{v}) = -\nabla p_k + \nu \nabla^2 \mathbf{v} + \frac{\rho \mathbf{g}}{\rho_0}, \quad (2.14)$$

where $p_k = \frac{p}{\rho_0}$ is the kinematic pressure that is usually used in incompressible CFD solvers. By removing the density from the buoyant force term, the Navier-Stokes equation can be further simplified. With the help of the thermal expansion coefficient β , the density can be expressed as a function of the temperature T , given as

$$\beta = -\frac{1}{\rho_0} \left(\frac{\partial \rho}{\partial T} \right)_p. \quad (2.15)$$

The derivative in Eq. (2.15) defines the rate of change of density with respect to temperature at constant pressure. By using linear approximation, the thermal expansion coefficient can be rewritten as

$$\beta \approx -\frac{1}{\rho_0} \frac{\rho - \rho_0}{T - T_0}, \quad (2.16)$$

where T_0 is the reference temperature. Equation (2.16) can be further simplified to:

$$\rho - \rho_0 \approx -\rho_0 \beta (T - T_0) \quad (2.17)$$

and finally

$$\frac{\rho}{\rho_0} \approx 1 - \beta (T - T_0). \quad (2.18)$$

The buoyant force term in Eq. (2.20) can now be expressed as

$$\frac{\rho \mathbf{g}}{\rho_0} \approx [1 - \beta (T - T_0)] \mathbf{g}, \quad (2.19)$$

where $\rho_k = \frac{\rho}{\rho_0} = [1 - \beta (T - T_0)]$ is the kinematic density used in the buoyant terms in the OpenFOAM code.

By setting the reference temperature T_0 before running the simulations and adding an equation for computing the temperature field, the Navier-Stokes equation can be expressed as

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} \otimes \mathbf{v}) = -\nabla p_k + \nu \nabla^2 \mathbf{v} + [1 - \beta(T - T_0)] \mathbf{g}. \quad (2.20)$$

As mentioned before, the Boussinesq approximation is valid for small density changes ($\nabla \rho \ll \rho_0$). The condition can be expressed as

$$\frac{\rho - \rho_0}{\rho_0} \ll 1, \quad (2.21)$$

which can be rewritten as

$$\beta(T - T_0) \ll 1. \quad (2.22)$$

When the condition in Eq. (2.22) is valid, density changes induced by temperature change are less than 1 %. For water that is being heated at room temperature, the density changes will be less than 1 % if the overall temperature change of the water is less than 2 °C ($T - T_0 < 2^\circ\text{C}$) [30].

2.3.5 Heat transfer

To account for the heat transfer of the flow and also to be able to express the density changes in the buoyant terms as a function of temperature, the temperature field needs to be computed.

The temperature equation can be derived from the energy equation which is based on the physical principle of energy being conserved. The total energy is given by $E = e + K$, as the sum of internal (e) and kinetic (K) energy. For an incompressible fluid, kinetic energy terms in the energy equation are negligible and the CFD solver is usually pressure-based which means that the internal energy equation is used. The scalar transport equation for internal energy (e) is defined as

$$\frac{\partial(\rho e)}{\partial t} + \nabla \cdot (\rho \mathbf{v} e) = -\nabla \cdot \mathbf{q} + S_e, \quad (2.23)$$

where S_e is a source term and \mathbf{q} is the local heat flux density vector [31]. With Fourier's

law of heat conduction, it is defined as

$$\mathbf{q} = -\kappa \nabla T. \quad (2.24)$$

κ is the material's thermal conductivity, which for water can be assumed to be constant since the medium is isotropic. ∇T is the temperature gradient. The internal energy is given by $e = c_p T$, where c_p is the specific heat capacity of water.

The internal energy equation can now be expressed as

$$\frac{\partial(\rho c_p T)}{\partial t} + \nabla \cdot (\rho \mathbf{v} c_p T) = -\nabla \cdot (\kappa \nabla T) + S_e. \quad (2.25)$$

The material properties ρ and c_p are assumed to be constant so they can be factored out, which gives the transport equation for temperature as

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{v} T) = \alpha \nabla^2 T + S_e, \quad (2.26)$$

where $\alpha = \frac{\kappa}{\rho c_p}$ is the thermal diffusivity of water. In turbulent flow, heat is (similarly to momentum) transported by the turbulent eddy motions. To account for the turbulent term in the temperature equation, the internal energy e is decomposed into an average and fluctuating component:

$$e(t) = \bar{e} + e'(t), \quad (2.27)$$

which is inserted in the energy equation in Eq. (2.23). The ensemble average of the energy equation gives rise to an additional turbulent heat flux term $\mathbf{q}_T = \overline{\rho v' e'}$ which is equivalent to the Reynolds stress term R_{ij} in the momentum equation [32]. Similarly to the Reynolds stress tensor (which was modeled according to the Boussinesq hypothesis with a turbulent eddy viscosity ν_T analogous to how the stress tensor τ_{ij} of a Newtonian fluid is related to the viscosity ν), the turbulent heat flux \mathbf{q}_T can be modeled using a turbulent thermal conductivity κ_T due to turbulent mixing. It is modeled by analogy with Fourier's law of heat conduction due to molecular interaction in Eq. (2.24) as

$$\mathbf{q}_T = -\kappa_T \nabla T. \quad (2.28)$$

By using an effective thermal conductivity $\kappa_{eff} = \kappa + \kappa_T$, the heat flux \mathbf{q} in Eq. (2.23) can be expressed with regard to the combined turbulent mixing and molecular interaction. Likewise, the thermal diffusivity in Eq. (2.26) can be expressed as the

effective thermal diffusivity defined as

$$\alpha_{eff} = \alpha + \alpha_T = \frac{\nu}{Pr} + \frac{\nu_T}{Pr_T}, \quad (2.29)$$

where Pr is the Prandtl number and Pr_T is the turbulent Prandtl number. The turbulent Prandtl number is given by

$$Pr_t = \rho c_p \frac{\nu_T}{\kappa_T}. \quad (2.30)$$

The final temperature equation in Eq. (2.26) can now after decomposition and averaging be expressed in tensor form as

$$\frac{\partial \bar{T}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{T} \bar{u}_j) = \frac{\partial}{\partial x_k} \left(\alpha_{eff} \frac{\partial \bar{T}}{\partial x_k} \right), \quad (2.31)$$

where the source term for radiation S_e has been eliminated. In general, heat transfer due to radiation in liquids is negligible in comparison to heat transfer due to convection and conduction.

2.3.6 CFD modeling in OpenFOAM

OpenFoam is an open-source CFD code written in C++, primarily developed by OpenCFD Ltd since 2004 [33]. The numerical method implemented in OpenFOAM uses the Finite Volume Method on unstructured meshes.

By using the Reynolds Stress modeling described in Eq. (2.5) together with the unsteady Reynolds Averaged Navier-Stokes equation in Eq. (2.4), the equation system that needs to be solved for a Newtonian, incompressible, and buoyancy-driven fluid of transient behaviour is defined in tensor form as

$$\begin{cases} \frac{\partial \bar{v}_i}{\partial x_i} = 0 \\ \frac{\partial \bar{v}_i}{\partial t} + \frac{\partial (\bar{v}_i \bar{v}_j)}{\partial x_j} = -\frac{\partial \bar{p}_k}{\partial x_i} + \frac{\partial}{\partial x_j} \left(2(\nu + \nu_T) \bar{S}_{ij} - \frac{2}{3} k \delta_{ij} \right) + [1 - \beta(\bar{T} - T_0)] g_i, \end{cases} \quad (2.32)$$

where the unknowns are the velocity components v_x, v_y, v_z and the pressure p .

The continuity equation (first equation) can be seen as a restriction on the velocity components in the momentum equation. That means that the velocity field solved in

the momentum equation must satisfy the continuity equation. The Reynolds stress term is given by $-\overline{v'_i v'_j} = -\frac{2}{3}k\delta_{ij} + 2\nu_T \overline{S_{ij}}$ and $[1 - \beta(\overline{T} - T_0)]g_i$ is the buoyant force term according to the Boussinesq approximation. Equation (2.32) shows the URANS equations for a buoyancy-driven flow of a viscous and incompressible fluid. It can be described as an expression for the momentum conservation (momentum equation) and there are three equations for each velocity component (v_x, v_y, v_z).

To numerically solve the URANS equations, the Finite Volume Method (FVM) described in section (2.3.1) is used to discretize the partial differential equations. The URANS equations are numerically integrated throughout the finite cell volumes in the computational domain. By decomposing the derivatives in the URANS/momentum equation by FVM, it can be reformulated in a general matrix form as

$$M\mathbf{v} = -\nabla p, \quad (2.33)$$

where M is a matrix of known coefficients of the semi-discretized momentum equation, v is the unknown velocity field and ∇p is the gradient of the unknown pressure field. The expression in Eq. (2.33) represents n linearized momentum equations, one for each cell centroid in the computational grid.

$$\begin{pmatrix} M_{1,1} & M_{1,2} & \dots & M_{1,n} \\ M_{2,1} & M_{2,2} & \dots & M_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n,1} & \dots & \dots & M_{n,n} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = - \begin{pmatrix} (\frac{\partial p}{\partial x_i})_1 \\ (\frac{\partial p}{\partial x_i})_2 \\ \vdots \\ (\frac{\partial p}{\partial x_i})_n \end{pmatrix} \quad (2.34)$$

The matrix M can be separated into diagonal and off-diagonal components, where A is a matrix containing the diagonal components of M :

$$A = \begin{pmatrix} A_{1,1} & 0 & \dots & 0 \\ 0 & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & A_{n,n} \end{pmatrix} \quad (2.35)$$

The momentum equation can be expressed as $M\mathbf{v} = A\mathbf{v} - H = -\nabla p$, where H is a residual vector defined as

$$H = A\mathbf{v} - M\mathbf{v}. \quad (2.36)$$

The H vector is needed to calculate the source term for the pressure equation. The discretized momentum equation can be expressed as

$$A\mathbf{v} - H = -\nabla p, \quad (2.37)$$

and together with the continuity equation, an equation for pressure can be derived. By setting the known boundary conditions, the URANS equations in Eq. (2.32) can be solved with CFD by using an iterative pressure-velocity coupling algorithm. This will give a solution which is a velocity field satisfying the continuity equation. A pressure-velocity coupling algorithm is a numerical procedure to solve the Navier-Stokes equations. In OpenFOAM, one can use either the SIMPLE, PISO, or PIMPLE algorithm. All of them use pressure-velocity coupling and they are initiated by solving the momentum equation in Eq. (2.33). They differ in how they loop over the equations, more specifically how many iterations are performed of the correction equations.

The SIMPLE algorithm

The SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm is widely used for steady-state simulations, where the flow is steady ($\frac{\partial \mathbf{v}}{\partial t} = 0$). The algorithm is initiated with a guess of the pressure field $p = p^*$, which is set as an initial condition. The discretized momentum equations in Eq. (2.33) are solved with the pressure set as the initial guess ($p = p^*$), and the velocity field \mathbf{v}^* is computed. To enforce mass conservation at each time step, the pressure needs to be corrected for the velocity field to satisfy the continuity equation ($\nabla \cdot \mathbf{v} = 0$).

The explicit pressure correction equation (Poisson equation) can be derived by explicitly solving the velocity from Eq. (2.37) as

$$\mathbf{v} = A^{-1}H - A^{-1}\nabla p, \quad (2.38)$$

where A is easily inverted since it is a diagonal matrix. By inserting the velocity expression into the continuity equation, the Poisson equation for pressure can be obtained as

$$\nabla \cdot (A^{-1}\nabla p) = \nabla \cdot (A^{-1}H), \quad (2.39)$$

where the matrix H depends on the velocity field. By inserting the previously computed velocity field \mathbf{v}^* into the pressure equation, the new corrected pressure p^{**} can be

computed. The pressure correction can be described as

$$p^{**} = p^* + p', \quad (2.40)$$

where p' is the pressure correction field. Now the new corrected velocity field \mathbf{v}^{**} can be calculated explicitly from Eq. (2.38) by using the corrected pressure p^{**} . The corrected velocity field \mathbf{v}^{**} now satisfies the discretized continuity equation, however, the pressure equation in Eq. (2.39) is no longer satisfied since the pressure field p^{**} is no longer correct and needs to be updated. The matrix H needs to be updated by recomputing Eq. (2.36). For the SIMPLE algorithm, this is done by solving the momentum equation all over again. Each time the momentum equation in Eq. (2.33) is solved, the algorithm enters a new iteration.

The steps of the SIMPLE algorithm can briefly be described as

1. Solve the momentum equation $M\mathbf{v} = -\nabla p$ for a predicted initial pressure p^* , compute the velocity field \mathbf{v}^* .
2. Compute the H matrix from $H = A\mathbf{v} - M\mathbf{v}$.
3. Correct the pressure with the Poisson equation for pressure, $\nabla \cdot (A^{-1}\nabla p) = \nabla \cdot (A^{-1}H)$. Solve the corrected pressure p^{**} .
4. Solve the corrected velocity field \mathbf{v}^{**} from $\mathbf{v} = A^{-1}H - A^{-1}\nabla p$.

The algorithm loops over steps 1 to 4 in so-called 'outer corrector' loops, where each loop represents an iteration of the steady-state simulation. Each iteration involves one predictor step (step 1) and one corrector step (steps 3 and 4).

The PISO algorithm

The PISO (Pressure-Implicit with Splitting of Operators) algorithm is used for unsteady (transient) flow and is seen as an extension of the SIMPLE algorithm. The PISO algorithm contains the same steps as the SIMPLE algorithm but performs an extra corrector step within a time step (outer loop iteration) instead of solving the momentum equation again (in the next outer loop iteration) as the SIMPLE algorithm does [34]. In the second corrector step the H matrix is updated and a second pressure correction and velocity correction is computed in steps 3 and 4.

The PISO solver performs one outer loop iteration (from steps 1 to 4) and usually two

to three inner loop iterations per time step, where it reloops over steps 2 to 4. The first reloop over steps 2 to 4 corresponds to the second corrector step. Within one time step the momentum equations in the predictor step are solved once, but the pressure field is at least twice-corrected so that:

$$p^{***} = p^{**} + p'' . \quad (2.41)$$

The velocity field will also be at least twice-corrected so that $v^{***} = v^{**} + v''$. In outer loop iterations, the non-linear and coupling terms are updated, whereas, in inner loops iterations the linear systems are updated [35]. This means that the PISO algorithm neglects to update the non-linear terms of the momentum equation during a time step.

A transient flow requires to be simulated with small time steps to achieve accurate results. By having small time steps, an improved diagonal dominance can be obtained (time derivatives are located in the diagonal) which implies that the equations will become more stable. In other words, a transient simulation is more stable than a steady-state [34]. Due to the time-stepping, the whole simulation will require many iterations though, which can make simulations quite time-consuming.

The CFL condition

The Courant–Friedrichs–Lewy (CFL) condition is a condition that is necessary for the convergence of explicit and semi-implicit schemes, to get an accurate solution. The condition is expressed in terms of the Courant number.

When using the PISO solver in OpenFOAM, a stability criterion needs to be fulfilled to keep the solution stable. The stability criterion states that the non-dimensional Courant number can not be larger than one ($Co \leq 1$). The Courant number in one-dimensional flow is defined as

$$Co = \frac{U \Delta t}{\Delta x}, \quad (2.42)$$

where U is the local cell velocity, Δt is the time step and Δx is the distance between cell centroids (cell size) [36]. In OpenFOAM, the calculation is based on the cell volume ΔV rather than the distance Δx . The Courant number can be explained as the fraction of a cell that the flow moves across in a time step. The Courant number provides a measure of the rate at which information is transported under the influence of a flux field. The Courant number criterion ($Co \leq 1$) states that the information from one cell

can only reach the next neighbor cell within a time step. In other words, it restricts the simulation from losing flow information in some cells.

The Courant number for an arbitrarily shaped three-dimensional cell is in OpenFOAM defined as

$$Co = \frac{1}{2} \Delta t \cdot \frac{\sum_f |\mathbf{U}_f \cdot \hat{\mathbf{n}}_f| A_f}{V_p}, \quad (2.43)$$

where $\sum_f |\mathbf{U}_f \cdot \hat{\mathbf{n}}_f|$ is the summation of the magnitude of the velocities normal to the face over all cell faces, \mathbf{U}_f is the face velocity, $\hat{\mathbf{n}}_f$ is the unit normal vector (pointing towards the cell centroid), A_f is the face area and V_p is the cell volume [37].

Pressure shift due to buoyancy

The total pressure in a fluid is defined as

$$p_0 = p + p_{dyn} + \rho gh, \quad (2.44)$$

where p is the static pressure, $p_{dyn} = \frac{1}{2} \rho \mathbf{v}^2$ is the dynamic pressure and ρgh is the hydrostatic pressure. h is the height which also can be defined as the coordinate z .

For buoyancy-driven flows, the buoyant term ρgh in the pressure equation needs to be modeled according to the Boussinesq approximation. To account for the buoyant term ρgh in the pressure correction of the pressure-velocity coupling algorithm, the alternative pressure p_{rgh} is used in the pressure calculations. In OpenFOAM, p_{rgh} is defined as the sum of static and dynamic pressure:

$$p_{rgh} = p_k - \rho_k gh, \quad (2.45)$$

where $p_k = \frac{p}{\rho_0}$ is the kinematic total pressure, ρ_k is the kinematic density and $\rho_k gh$ is the hydrostatic pressure. So when calculating the pressure gradient $-\nabla p_k$ in the momentum equation, the buoyant term $\rho_k gh$ can be accounted for in the z direction (where $\rho_k = [1 - \beta(\bar{T} - T_0)]$ according to the Boussinesq approximation). The z component of the pressure gradient is computed as

$$-\frac{\partial p_k}{\partial z} = -\frac{\partial p_{rgh}}{\partial z} - \rho_k g_i - g_i x_i \frac{\partial \rho_k}{\partial z}, \quad (2.46)$$

where x_i is the i^{th} component of the position vector \mathbf{x} (the scalar product $g_i x_i$ equals to

gh). The averaged pressure gradient used in the momentum equation is defined as

$$-\frac{\partial \bar{p}_k}{\partial x_i} = -\frac{\partial \bar{p}_{rgh}}{\partial x_i} - \rho_k g_i - g_i x_i \frac{\partial \rho_k}{\partial z}. \quad (2.47)$$

The Reynolds-Averaged momentum equation can thus be rewritten as

$$\frac{\partial \bar{v}_i}{\partial t} + \frac{\partial (\bar{v}_i \bar{v}_j)}{\partial x_j} = -\frac{\partial \bar{p}_{rgh}}{\partial x_i} - g_i x_i \frac{\partial \rho_k}{\partial z} + \frac{\partial}{\partial x_j} \left(2(\nu + \nu_T) \bar{S}_{ij} - \frac{2}{3} k \delta_{ij} \right), \quad (2.48)$$

where the two terms $\rho_k g_i$ have cancelled each other out ($-\rho_k g_i + \rho_k g_i = 0$). Eq. (2.48) is the final Reynolds-Averaged momentum equation defined in the OpenFOAM code.

Governing equations

This section will present the governing equations (based on how they have been defined in the OpenFOAM source code) for a buoyancy-driven flow of a viscous and incompressible fluid for which the Boussinesq approximation is valid. The equations are based on the OpenFOAM solver used, the *buoyantBoussinesqPimpleFoam* solver. It is a transient solver for a buoyant, turbulent flow of incompressible fluids, with optional mesh motion and mesh topology changes [38].

The unsteady Reynolds-Averaged Navier-Stokes equations are defined with respect to the pressure shift as in Eq. (2.48):

$$\frac{\partial \bar{v}_i}{\partial t} + \frac{\partial (\bar{v}_i \bar{v}_j)}{\partial x_j} = -\frac{\partial \bar{p}_{rgh}}{\partial x_i} - g_i h \frac{\partial \rho_k}{\partial z} + \frac{\partial}{\partial x_j} \left(2(\nu + \nu_T) \bar{S}_{ij} - \frac{2}{3} k \delta_{ij} \right).$$

They are solved together with the continuity restriction which is defined in Eq. (2.32) as

$$\frac{\partial \bar{v}_i}{\partial x_i} = 0.$$

The kinematic pressure \bar{p}_{rgh} is computed in Eq. (2.45) as

$$p_{rgh} = p_k - \rho_k g h,$$

and this is the pressure that is corrected in the pressure equation. The kinematic density ρ_k is defined according to the Boussinesq approximation as

$$\rho_k = 1 - \beta(\bar{T} - T_0). \quad (2.49)$$

The temperature field T is determined by the transport equation for temperature as in Eq. (2.31):

$$\frac{\partial \bar{T}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{T} \bar{u}_j) = \frac{\partial}{\partial x_k} \left(\alpha_{eff} \frac{\partial \bar{T}}{\partial x_k} \right).$$

The $k - \epsilon$ turbulence model is used together with the Boussinesq eddy viscosity hypothesis to close the system of equations. The transport equation for the turbulent property k is defined in OpenFOAM as

$$\rho \frac{\partial k}{\partial t} + \frac{\partial}{\partial x_i} (\rho k \bar{v}_i) = \frac{\partial}{\partial x_j} \left[\rho \left(\nu + \frac{\nu_T}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + \rho \nu_T 2 \bar{S}_{ij} \bar{S}_{ij} - \nu_T g_i \frac{\partial \rho_k}{\partial x_i} - \rho \epsilon + S_k, \quad (2.50)$$

where $P_k = \nu_T 2 \bar{S}_{ij} \bar{S}_{ij}$ represents the transport of turbulent kinetic energy k due to production and $P_b = -\nu_T g_i \frac{\partial \rho_k}{\partial x_i}$ represents the effect of buoyancy. By considering the Boussinesq approximation, the buoyant term can be expressed as $P_b = \nu_T g_3 \beta \frac{\partial \bar{T}}{\partial z}$, where g_3 is the gravitational acceleration.

The transport equation for the turbulent property ϵ is defined in OpenFOAM as

$$\rho \frac{\partial \epsilon}{\partial t} + \frac{\partial}{\partial x_i} (\rho \epsilon \bar{v}_i) = \frac{\partial}{\partial x_j} \left[\rho \left(\nu + \frac{\nu_T}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] + \rho C_1 \frac{\epsilon}{k} \nu_T 2 \bar{S}_{ij} \bar{S}_{ij} - C_2 \rho \frac{\epsilon^2}{k} + S_\epsilon, \quad (2.51)$$

where the non-dimensional constants C_1 and C_2 are set to 1.44 and 1.92 respectively.

With the Boussinesq eddy viscosity hypothesis, the turbulent viscosity is modeled as

$$\nu_T = \frac{C_\mu k^2}{\epsilon}, \quad (2.52)$$

where the non-dimensional constant C_μ is set to 0.09.

2.3.7 Boundary layer treatment

Due to the no-slip boundary condition (zero velocity at the wall), the fluid flowing along a solid surface forms a boundary layer adjacent to the wall. The boundary layer is characterized by an increasing streamwise velocity in the direction perpendicular to the surface, where the velocity takes its maximum value (free stream velocity v_∞) far above the wall. The boundary layer is a small region of the fluid where viscous forces are dominating due to the friction effects from the surface. The friction effects at the wall give rise to viscous stresses in the vicinity of the wall. Depending on Reynolds number,

the flow within the boundary layer can be either laminar or turbulent. The boundary layer thickness depends on the flow parameters and whether the flow is developing or is fully developed. The relation between the viscous shear stress and the velocity gradient is given as

$$\tau = \mu \frac{\partial U}{\partial y}, \quad (2.53)$$

where U is the fluid velocity parallel to the wall and y is the coordinate perpendicular to the wall. In CFD codes based on the 2^{nd} order Finite Volume method, the velocity, and other mean flow properties are assumed to vary linearly across the cell. However, the velocity gradient is steeper in turbulent boundary layers and the velocity profile is non-linear, which has been proven by DNS data of fully-developed turbulent channel flows [39]. See Appendix A.1 for a graph of the universal mean velocity profile (red graph) of a boundary layer belonging to a fully-developed turbulent channel flow, according to the law of the wall.

The dimensionless wall coordinate y^+ is given by

$$y^+ = \frac{y u_\tau}{\nu}, \quad (2.54)$$

where y is the local distance normal to the wall and u_τ is the friction velocity defined as

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}. \quad (2.55)$$

The y^+ value can be interpreted as a local Reynolds number, a measure of distance normal to the wall that scales with the boundary layer thickness. Its magnitude explains the relative importance between viscous and turbulent processes. It is usually taken into consideration when generating the mesh since it can be used to decide how fine the mesh cells should be (the first cell height). To resolve the flow near the wall, it is usually required to have $y^+ \leq 1$.

The dimensionless velocity u^+ is defined as the parallel velocity divided by the friction velocity:

$$u^+ = \frac{U}{u_\tau}. \quad (2.56)$$

According to the law of the wall, u^+ relates to y^+ ($u^+ = f(y^+)$) in different ways depending on which region of the boundary layer [20]. The different layers of a boundary layer are defined as:

- Viscous sub-layer, $y^+ < 5$ $u^+ = y^+$
- Buffer layer, $5 < y^+ < 30$
- Log-law region, $y^+ > 30$ $u^+ = \frac{1}{\kappa} \ln(y^+) + B$

κ is the von Karman's constant ($= 0.41$) and B is a constant which is equal to 5.2. Viscous forces are dominating in the viscous sub-layer and here the wall shear stress is almost entirely viscous. In the buffer region, turbulence becomes significant. In the log-law region, inertial (turbulent) forces are dominating and the logarithmic law of the wall is used to describe the velocity profile.

Wall shear stress

To resolve the flow in turbulent boundary layers accurately, one needs to use a fine mesh resolution close to the wall to resolve the steep velocity gradient accurately. A piece-wise linear function will then be used to model the velocity gradient. The first cell center should then be located within the viscous sub-layer so that $y^+ < 5$.

Another option for resolving the boundary layer, which allows using a coarse mesh (larger cells) close to the wall, is to model the kinematic viscosity ν with a non-linear function. By correcting the near wall kinematic viscosity ν_w , the correct wall shear stress τ_w can be computed even though the resolved velocity gradient is not accurate. This can be done by introducing a turbulent viscosity ν_T in the log-law region of the boundary layer, which can be done by using wall functions. When using wall functions, the first cell center should be placed in the log-law region.

From Eq. (2.53) the wall shear stress can be expressed as

$$\frac{\tau_w}{\rho} = \nu_w \frac{\Delta U}{\Delta y} = \nu_w \frac{U_p}{y_p}, \quad (2.57)$$

where U_p is the local velocity in the first cell centroid (cell adjacent to the wall) and y_p is the normal distance from the wall to the cell centroid.

The near-wall kinematic viscosity is modeled as a sum of the laminar viscosity which is a material property and the turbulent viscosity ($\nu_w = \nu + \nu_T$). With wall functions,

the turbulent viscosity ν_T is modeled as

$$\nu_T = \begin{cases} 0 & y^+ < 11.25 \\ \nu \left(\frac{y^+ \kappa}{\ln(Ey^+)} - 1 \right) & y^+ > 11.25, \end{cases} \quad (2.58)$$

where E is a wall roughness parameter ($= 9.8$ for smooth walls) [35]. $y^+ = 11.25$ is where the wall function for the laminar viscous sub-layer and the wall function for the log-law region intersect. In the viscous sub-layer, the flow is assumed to be laminar and flow properties are assumed to vary linearly ($\nu_w = \nu$). In the log-law region, the fluid is turbulent and flow properties can no longer be assumed to vary linearly. Therefore, the near wall kinematic viscosity is corrected within this region by adding a turbulent viscosity ν_T to the near wall kinematic viscosity ($\nu_w = \nu + \nu_T$).

The wall shear stress can now be modeled as

$$\tau_w = \begin{cases} \mu \frac{U_p}{y_p} & y^+ < 11.25 \\ \mu \frac{y^+ \kappa}{\ln(Ey^+)} \frac{U_p}{y_p} & y^+ > 11.25. \end{cases} \quad (2.59)$$

The CFD code will automatically switch between the different calculations of the turbulent viscosity ν_T (depending on the current y^+ value) to ensure that the right wall shear stress is computed. In the buffer region, a blending function can be used to calculate the turbulent viscosity. The method of using wall functions allows the CFD code to use the same calculation of the velocity gradient ($\frac{\partial U}{\partial y}$) across the domain [40].

The wall function *nutkWallFunction* in OpenFOAM provides a turbulent viscosity condition based on the turbulent kinetic energy (k). That means that the calculation of the velocity scale u_τ is based on the turbulent kinetic energy rather than the wall shear stress τ_w as in Eq. (2.55). That introduces another dimensionless distance used in the wall function, called y^* , which is defined as

$$y^* = C_\mu^{1/4} y \frac{\sqrt{k}}{\nu}, \quad (2.60)$$

where the velocity scale is given as $u_\tau = C_\mu^{1/4} \sqrt{k}$. The y^* value plays the same role as y^+ in measuring a dimensionless distance normal to the wall for the mesh cells. The difference is that y^* is less computationally expensive since no iteration is required (for

the wall shear stress) as it is when using y^+ . The turbulent kinetic energy can be used to directly compute y^* which then is used to directly compute the wall shear stress. This is appropriate for more general flows which may have stagnation points and separation [41].

Wall heat transfer

Similar to velocity profiles, the near-wall temperature profile needs to be modeled. To be able to use large cells near the wall, the non-linear variation of temperature between the first cell centroid and the wall is modeled. It can be modeled in a similar way as the velocity, by fitting empirical functions to the data (one for the viscous sub-layer and one for the log-law region).

The dimensionless (normalized) temperature T^* is defined as

$$T^* = \frac{(T_w - T)\rho c_p u_\tau}{q_w}, \quad (2.61)$$

where T_w is the wall temperature, q_w is the wall heat flux and c_p is the specific heat constant. u_τ is the friction velocity defined as

$$u_\tau = C_\mu^{1/4} \sqrt{k}. \quad (2.62)$$

The dimensionless wall coordinate y^* normal to the wall is defined as

$$y^* = \frac{y u_\tau}{\nu}. \quad (2.63)$$

With the standard wall function formulation, the dimensionless temperature is calculated as

$$T^* = Pr \ y^* \quad y^* < 5 \quad (2.64)$$

$$T^* = Pr_T \left(\frac{1}{\kappa} \log(E y^*) + P \right) \quad 30 < y^* < 200, \quad (2.65)$$

where $Pr = \frac{\nu}{\alpha}$ is the molecular Prandtl number and Pr_T is the turbulent Prandtl number which is treated as a constant (0.85). P is another empirical function used to translate the log-law curve. According to Jayatilke (1969) [42] the P quantity can be determined from experimental data as

$$P = 9.24 \left[\left(\frac{Pr}{Pr_T} \right)^{3/4} - 1 \right] \left[1 + 0.28 e^{-0.007(Pr/Pr_T)} \right]. \quad (2.66)$$

The intersection point y_L^* of the two empirical wall functions in Eq. (2.64) and (2.65) is dependent on the Prandtl number. Therefore, the CFD code needs to compute the intersection point to switch between the functions. For water, the intersection point is $y^* = 7.6$.

The aim when simulating a heat transfer case with CFD is to predict the wall heat transfer q_w . The temperature at the wall T_w and the temperature at the cell center of the first cell adjacent to the wall (T_p) are known. If the first cell is located in the viscous sub-layer, the temperature variation from the wall to the cell centroid can be assumed to be linear. With Fourier's law, the wall heat transfer in the viscous sub-layer ($y^* < y_L^*$) can then be calculated as

$$q_w = -\kappa \frac{\partial T}{\partial y} \bigg|_{y=0} = \rho c_p \alpha \frac{T_w - T_p}{y_p}. \quad (2.67)$$

For the log-law region, Eq. (2.61) can be rearranged to find q_w .

$$q_w = \frac{(T_w - T) \rho c_p u_\tau}{T^*} \quad (2.68)$$

By inserting the expression for T^* in Eq. (2.65) into Eq. (2.68) the wall heat flux q_w can be calculated as

$$q_w = \frac{(T_w - T) \rho c_p u_\tau}{Pr_T \left(\frac{1}{\kappa} \log(Ey^*) + P \right)}. \quad (2.69)$$

Since one wants to predict the same wall heat flux regardless of in what region of the boundary layer, the two expressions of q_w in Eq. (2.67) and (2.69) are set equal to each other.

$$\rho c_p \alpha \frac{T_w - T_p}{y_p} = \frac{(T_w - T) \rho c_p u_\tau}{Pr_T \left(\frac{1}{\kappa} \log(Ey^*) + P \right)} \quad (2.70)$$

By simplifying and rearranging, the near wall thermal diffusivity α_w can be defined as

$$\alpha_w = \frac{u_\tau y_p}{Pr_T \left(\frac{1}{\kappa} \log(Ey^*) + P \right)}. \quad (2.71)$$

One will get the same wall heat flux in the log-law region (as in the viscous sub-layer) if the thermal diffusivity in the log-law region ($y^* > y_L^*$) is calculated as in Eq. (2.71).

Finally, the near-wall thermal diffusivity can be modeled as

$$\alpha_w = \begin{cases} \alpha & y^* < y_L^* \\ \frac{u_\tau y_p}{Pr_T \left(\frac{1}{\kappa} \log(Ey^*) + P \right)} & y^* > y_L^* \end{cases} \quad (2.72)$$

2.4 The reservoir

The reservoir, which is to be modeled with CFD, is located in the distribution system of Norrvatten and has the function of maintaining flow equalization in the pipeline network by storing drinking water. The volume in the reservoirs should partly cover diurnal variations in consumption of water, and also form a certain reserve.

2.4.1 Operational conditions

During periods of low consumption, more water is stored in the tank, which then is distributed during periods of high consumption. Thus, the filling and draining of the tank completely depend on certain circumstances such as the consumers' water consumption and the pressure distribution in the supply network. However, the operational strategies do not vary radically under normal conditions, and the typical operational strategies can therefore be summarized in the following points:

- At night during a period ranging from 01:00 to 09:00, the reservoir is filled with water. The water level increases from 2-4 meters to 6-9 meters.
- The reservoir is kept closed for a few hours before it is emptied.
- The reservoir is emptied during a period ranging from 15:00 to 22:00, to a minimum of 1 metres.
- The reservoir is closed a few hours before it is filled again.

During one day the reservoir is usually filled once and drained once. The volumetric flow rates of incoming and outgoing water vary between 0.01-0.15 m³/s.

2.4.2 Water sampling

The reservoir has an external sampling tap located in a pipeline where both in- and outgoing water flows. During a filling process, there is incoming water in the pipe, and during a draining process, the pipe contains water leaving the reservoir. From the sampling tap, temperature is measured with a temperature gauge. The BactoSense flow cytometer is also located here and it measures the water level and microbial activity. Due to the location of the sampling tap, measurements of the outgoing water can only be taken during a draining process. During the filling process, measurements will instead be taken of the incoming water which is not of interest.

Apart from the flow cytometer analysis, water samples are also taken from the sampling tap to ensure that the drinking water meets the requirements of the Swedish Food Agency. The water samples are analyzed in an accredited laboratory by heterotrophic plate counting, which is described in section 2.1.2. Microbial cultivation is performed for both three and seven days.

2.4.3 Water level

The water level is a variable that is completely determined by the operational conditions, and it can be used as a tool for analyzing and comparing different operational conditions of the reservoir. See Appendix A.2 for an example of how the water level varies in the reservoir during seven days. Figure 2.4.1 shows the daily variation of the water level for each day of the seven-day period 16th to 23rd September 2021.

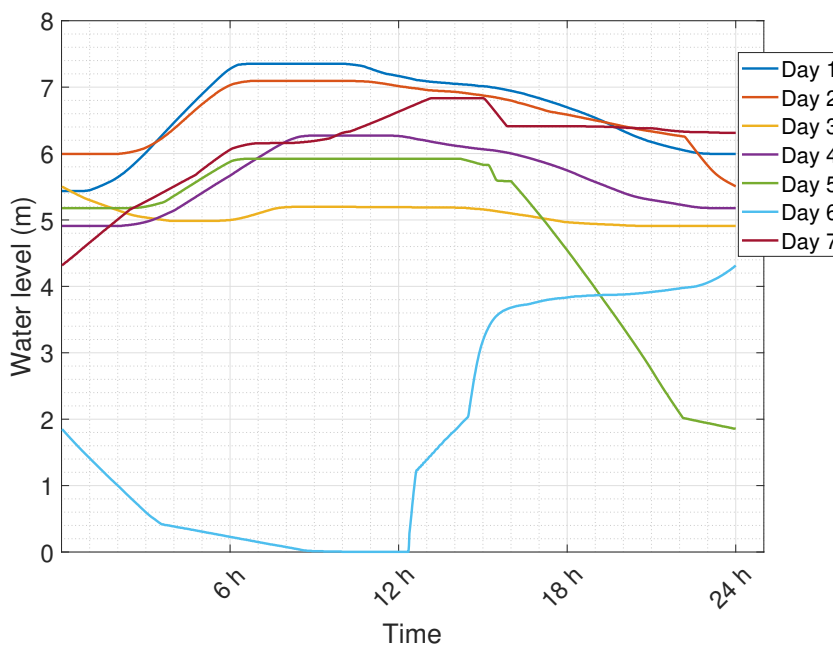


Figure 2.4.1: The daily variation of the water level in the reservoir during the period 16th to 23rd September 2021.

The water level during day 6 (light blue line) shows that there was a complete draining of the reservoir around the time 12:00. The other curves in the graph show the typical trend for the daily variation of the water level. For that actual week, the typical daily inflow period results in a water level increase of up to two meters in the early hours

of the morning. The filling process may last three to six hours. The typical draining process starts around noon or later in the afternoon and proceeds for around 12 hours.

2.4.4 Temperature

The temperature of the water in the reservoir is a physical quantity that is determined by external factors such as the outside air temperature and the temperature of the water in the water distribution system. It is nevertheless interesting to analyze the temperature distribution in the reservoir to examine the thermal stratification. The temperature is measured by Norrvatten with a thermometer located in the sampling tap pipe. During the filling process, the temperature registered is the one of the incoming water, and during the draining process, it is the one of the outgoing water. There are no thermometers located inside the reservoir. Figure 2.4.2 shows the measured temperature and water level variation during seven days.

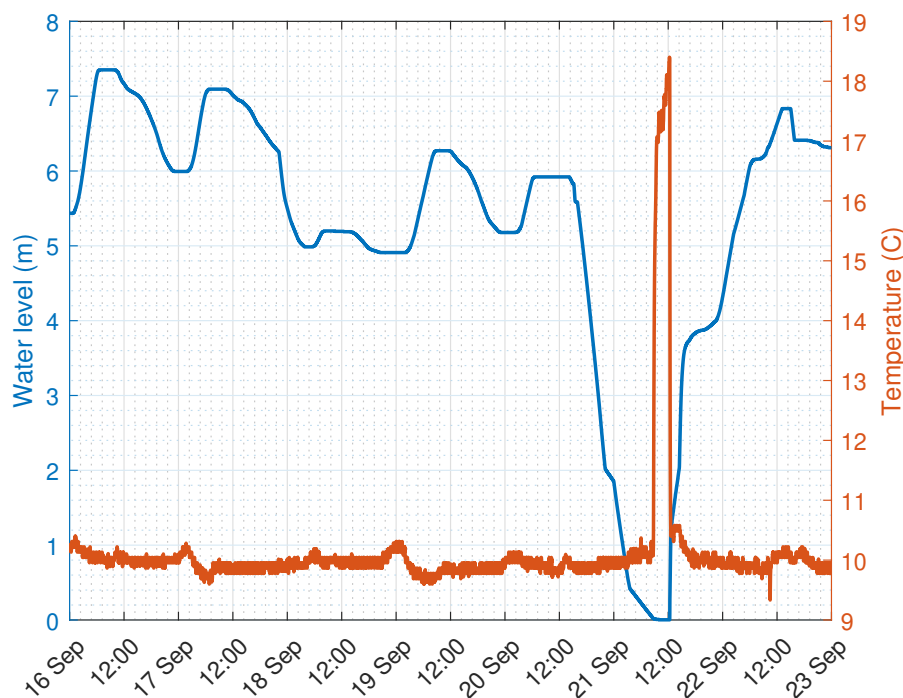


Figure 2.4.2: The temperature and water level as a function of time in the reservoir during the period 16th to 23rd September 2021.

The temperature has small fluctuations around 10 °C with an exception for the distinct peak on the 21st September at 12:00. This radical increase in temperature (to 18 °C) is due to the complete draining of the reservoir which results in the small remaining volume of water, left at the bottom, being quickly heated by the surrounding air and the

bottom wall in the reservoir. It can also be seen that there are some small temperature drops at the beginning of the filling processes of the reservoir. This can be explained by the fact that the incoming water has a lower temperature than the already existing water in the reservoir/pipe.

2.4.5 Microbiological measurements of the reservoir water

The microbial activity of the water is measured with the Bactosense flow cytometer in the same pipe the temperature is measured. Since the same pipe is used for in- and outgoing water the flows are separated by not having the inlet and outlet open at the same time [4]. Hence, during the filling process the BactoSense is measuring the microbial activity of the incoming water, and vice versa.

Figure 2.4.3 shows the cyclic change of the water level and the HNA bacteria proportion (of total intact bacteria (ICC)) in the reservoir during four representative days [4].

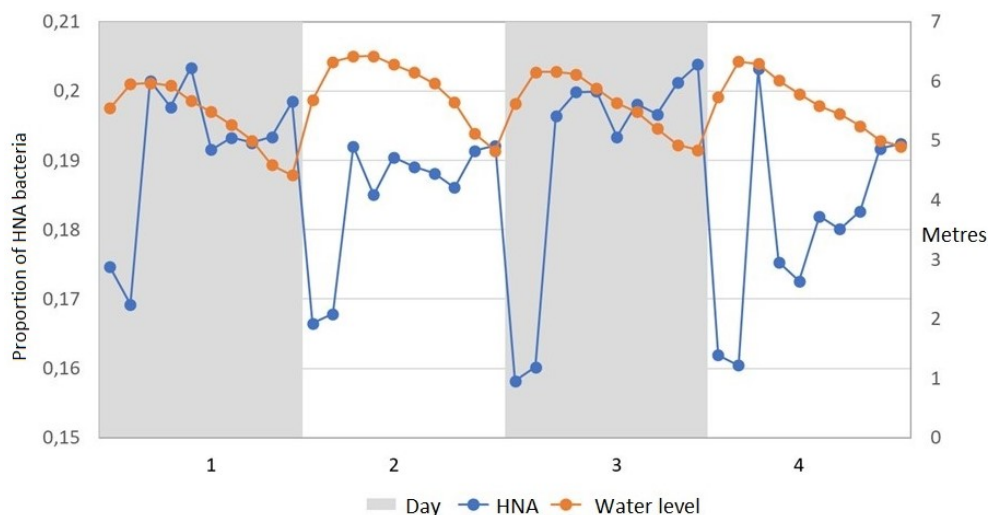


Figure 2.4.3: The cyclical change of the water level and the HNA proportion (of ICC) in the reservoir during four representative days.

It can be seen that the HNA proportion follows a certain cycle and decreases during the night when the reservoir is being filled. This can be explained by the fact that fresh incoming water has a lower bacteria regrowth compared to the volume of water that has been staying in the reservoir for a while. The HNA percentage of the incoming water is around 16-17 percent. The HNA percentage indicates the proportion of HNA bacterial cells of the total bacterial cells in a milliliter of water.

Figure 2.4.4 shows the water level variation (grey line) and the measured HNA

percentage (blue line) for the time interval June 2020 to December 2021 [4]. The dips of the grey line in the graph represent the complete drainings of the reservoir.

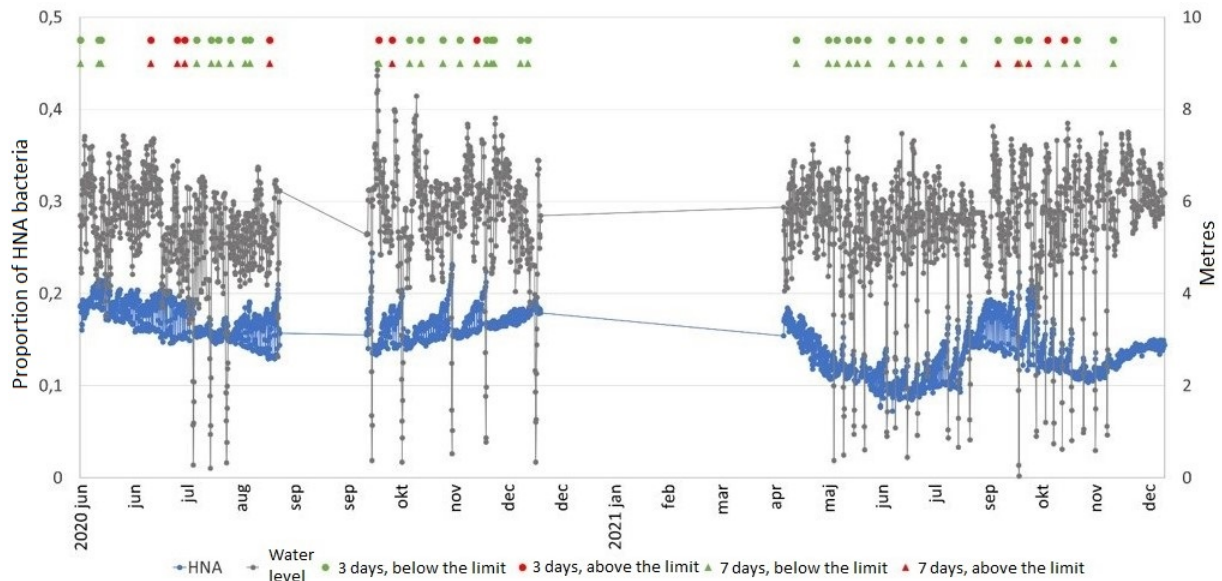


Figure 2.4.4: Measurements of the water level and HNA percentage in the reservoir from June 2020 to December 2021, including cultivation results.

In Figure 2.4.4 cultivation results are shown in the upper part of the graph. Circular markings represent heterotrophic bacteria and triangular markings represent slow-growing bacteria. Green circles/triangles define that cultivation results are approved, and the red color demonstrates that samples exceed the limit for the microbiological analyses according to regulations of the Swedish Food Agency [4][6].

Figures 2.4.5 and 2.4.6 show the change in the HNA percentage with respect to the water level and temperature, respectively. The data is for the seven-day period (16-22 Sep 2021).

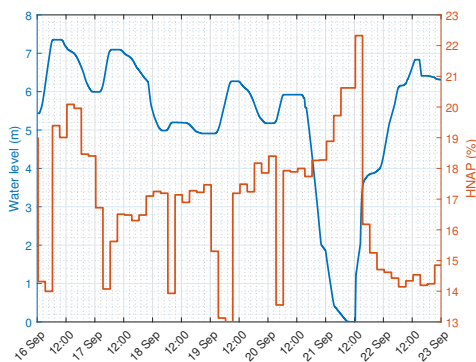


Figure 2.4.5: The HNA percentage and water level as a function of time in the reservoir.

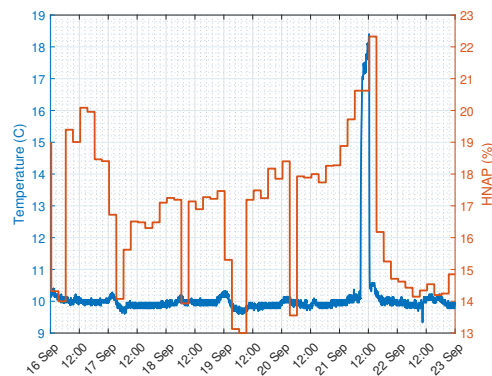


Figure 2.4.6: The HNA percentage and temperature as a function of time in the reservoir

Figure 2.4.7 shows the variation of the total cell count (TCC), HNA content (HNAC), and HNA percentage (HNAP) per ml water, with respect to time.

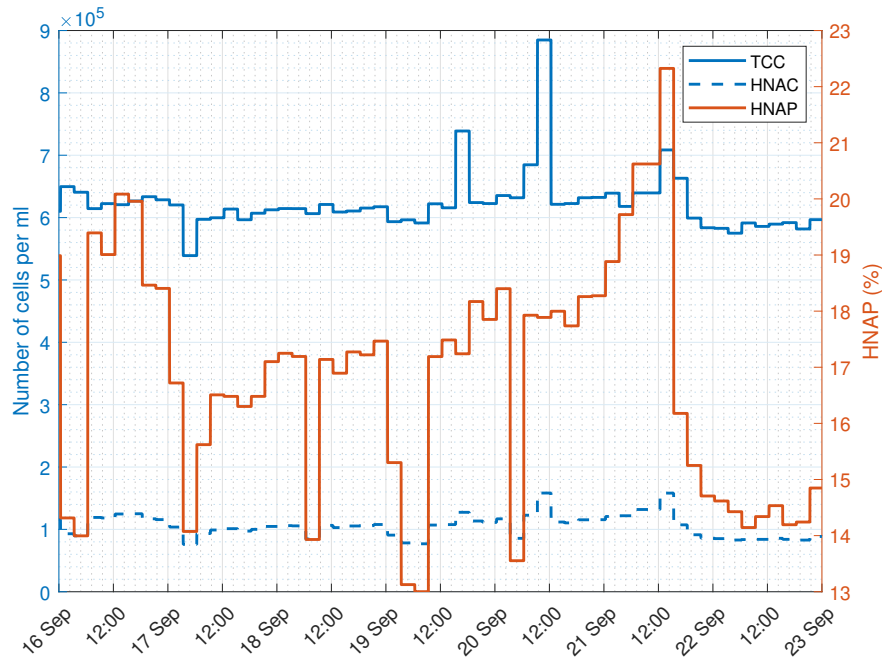


Figure 2.4.7: TCC, HNAC, and HNAP per ml water as a function of time.

It can be seen that the content of high nucleic acid bacteria corresponds to around 14-22 percent of the total cell count in one milliliter of water, and during the seven days it varies a lot probably due to the variation in water level. Figure 2.4.7 shows that the graph of the HNA bacteria content approximately follows the same trend as the one of the total cell count which means that the increase of HNA bacteria is fairly relative to the increase of total bacteria. However, at some times the HNAP graph seems to change independently of the TCC graph. This indicates that there might be other factors that have an impact on the change of the HNA bacteria proportion (without impacting the total cell count).

Operational measures

Norrvatten takes some operational measures to inhibit the regrowth of bacteria in the reservoir. Once in a while during certain periods such as late summer and fall, the reservoir is completely emptied to avoid marks on the water quality. During May-November 2021, the reservoir was completely emptied 20 times, as can be seen in Figure 2.4.4. This operational method contributed to a decreased proportion of HNA

(high nucleic acid) bacteria in the water that was sampled a while after the draining. An explanation for this may be that the entire surface film or large parts of it (where the majority of bacteria reside) had been removed when the reservoir was completely drained.

Another way to get rid of the surface film is to overflow the reservoir. The excess water will then be discharged as stormwater and led to a storm ditch.

Chapter 3

Methods

3.1 CFD setup specification in OpenFOAM

This section will describe the procedure for setting up the CFD simulations of the reservoir model in OpenFOAM.

3.1.1 Physical setup

When setting up the simulations to solve the RANS equations, the following physical assumptions were made.

The fluid flow is:

- Newtonian and three-dimensional.
- Incompressible: density variations are negligible. Can be applied when the Mach number is below 0.3.
- Turbulent: fluid undergoes irregular fluctuations or mixing.
- Transient (unsteady): flow properties are dependent on time.

Apart from the model specifications above, the flow is modeled as a single-phase flow where the material is liquid water and the whole domain consists of the reservoir water volume. The water surface is modeled as a rigid lid (a wall that moves vertically as the reservoir is being filled/drained). The flow is buoyancy-driven and the Boussinesq approximation is used, meaning that the density variations are neglected in all terms except in the buoyancy term.

Table 3.1.1 shows the transport properties specifications set in OpenFOAM. The properties of water are set for a reference temperature of 10.2 °C (283.35 K) except for the thermal expansion coefficient (β) which is set for a temperature of 17 °C. The density of water (which is temperature dependent) will vary between approximately 999.57 to 999.8 kg/m³.

Table 3.1.1: Transport properties of water.

Kinematic viscosity (m^2/s)	Thermal expansion coefficient ($1/K$)	Reference temperature (K)	Prandtl number
1.3e-6	0.88e-4	283.35	7.56

3.1.2 The reservoir model

The geometry of the reservoir model was created with the *blockMesh* utility in OpenFOAM according to the dimensions of the reservoir. The model is three-dimensional and has an in- and outlet located in accordance with the dimensions of the reservoir, see Figure 3.1.1. The computational domain consists of the water volume in the reservoir and expands vertically as the water level rises.

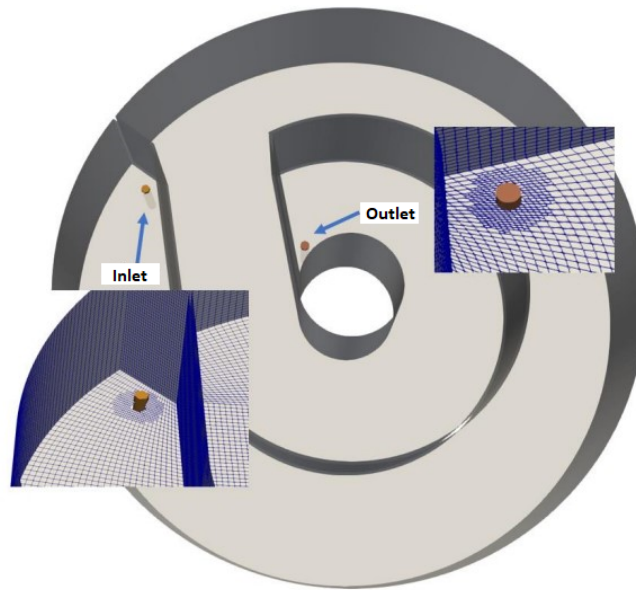


Figure 3.1.1: Simplified model of the reservoir with an enlarged view of the in- and outlet.

3.1.3 Mesh generation process

The computational grid was created with the mesh generation utility `blockMesh` and the mesh was generated from a dictionary file called *blockMeshDict*. In this file, the domain geometry was decomposed into three-dimensional hexahedral blocks (see Appendix B.1.1 to view the code from the *blockMeshDict* file).

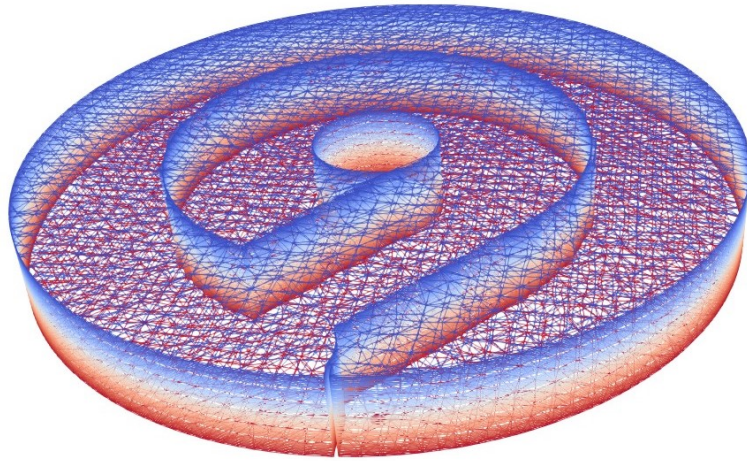


Figure 3.1.2: Visualization of the edges of the surface mesh of the reservoir. The color indicates the pressure distribution.

Each block is defined by 8 vertices, one at each corner of the hexahedral block. The vertices of the blocks are defined in a list called *vertices* in the file. The edges of the blocks are straight lines or arcs. Each edge joins two vertex points. All the edges which are not straight are defined in a list called *edges* in the *blockMeshDict* file. The blocks are defined in a list called *blocks*. The number of cells in each direction of the block is specified here. The cell expansion ratio in each direction of the block can as well be specified here. The boundary faces of the geometry (such as the vertical walls, floor, and water level) are also defined in the *blockMeshDict* file.

To apply boundary conditions, mesh boundaries need to be defined. A boundary is divided into a set of patches, where each patch consists of a set of boundary faces.

Dynamic mesh

To simulate the reservoir as a single-phase flow and let the water volume form the entire computational domain, the mesh needs to be dynamic. As the water level patch is moving up or down due to in or outflow, the mesh needs to be expanding or shrinking consistently in the vertical direction.

In the *dynamicMeshDict* file located in the *constant* directory, the diffusivity of the mesh motion was defined to be uniform. That means that the motion of the water level patch is uniformly diffused into the domain. The moving boundary that controls the mesh motion is the water level patch. Its motion is defined in the *pointMotionUz* file located in the 0 directory.

Inlet and outlet

The inlet and outlet patches are created with the *topoSet* utility, which operates through the dictionary file *topoSetDict* located in the *system* directory (see Appendix B.1.2).

The patches were created by a few *topoSet* actions defined in the *topoSetDict* file. The approach for creating the inlet and outlet patch can be described as:

1. A cell set named *c0* was created by selecting all the cells in a cylinder. The cylinder geometry is defined by specifying the coordinates of the endpoints of the cylinder (*p1* and *p2*) and specifying the radius (which is set to 0.3 meters).
2. A face set was created by selecting all faces of the cells in the cell set *c0*.
3. A subset (named *inletFaces*) of the previously created face set was created by selecting all the faces that are a part of the "walls_floor" patch.

Finally, a face set with all its faces located in line with the floor of the reservoir has been created. Figures 3.1.3 and 3.1.4 show the inlet and outlet patch respectively and the refined mesh surrounding the patches. The radius of the refinement area (of inlet and outlet) is set to 1 meter.

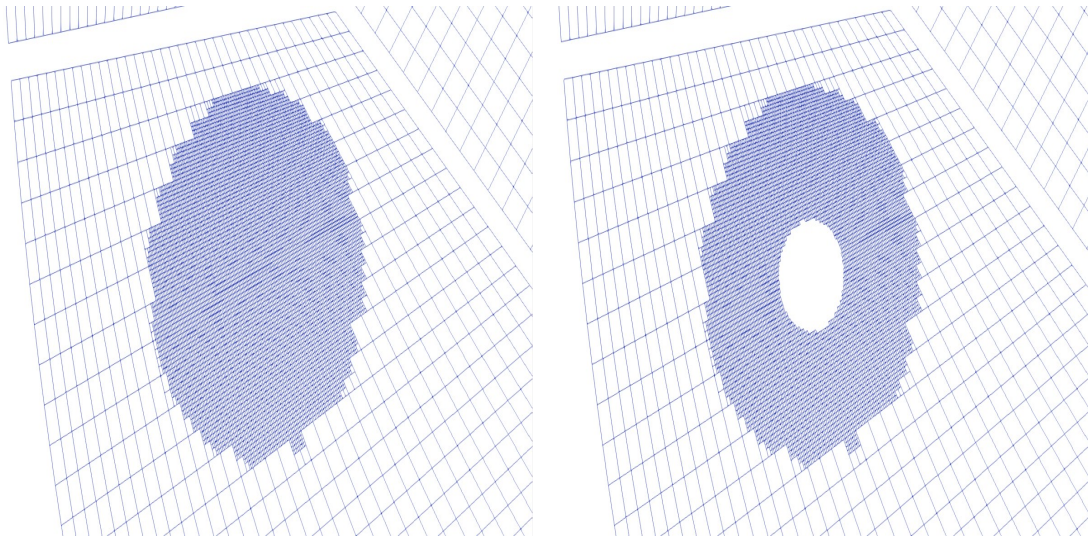


Figure 3.1.3: View of the inlet seen from above the reservoir with only the floor of the reservoir visible.

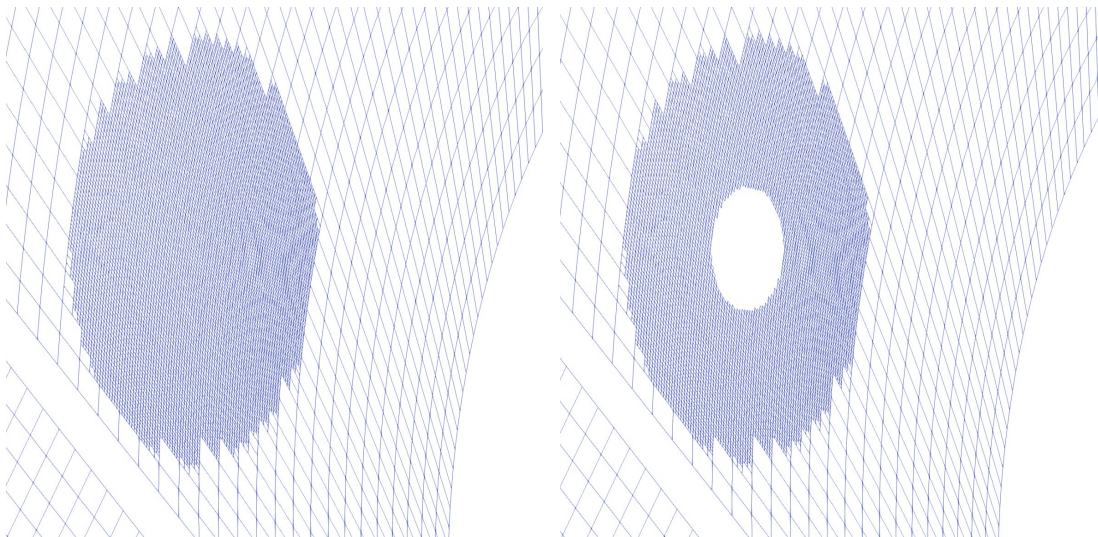


Figure 3.1.4: View of the outlet seen from above the reservoir with only the floor of the reservoir visible.

Vent

The "vent" is a patch that was created later on, to ensure that the law of conservation of mass is valid in the reservoir. The water level patch should move vertically with regard to the inflow or outflow of water. But since the motion of the water level needs to be manually specified, it is independent of the inflow and outflow rate. Therefore, a vent (hole) was created at the bottom of the reservoir through which water can either flow out (leave the reservoir) or flow in (enter the reservoir). See Figure 3.1.5.

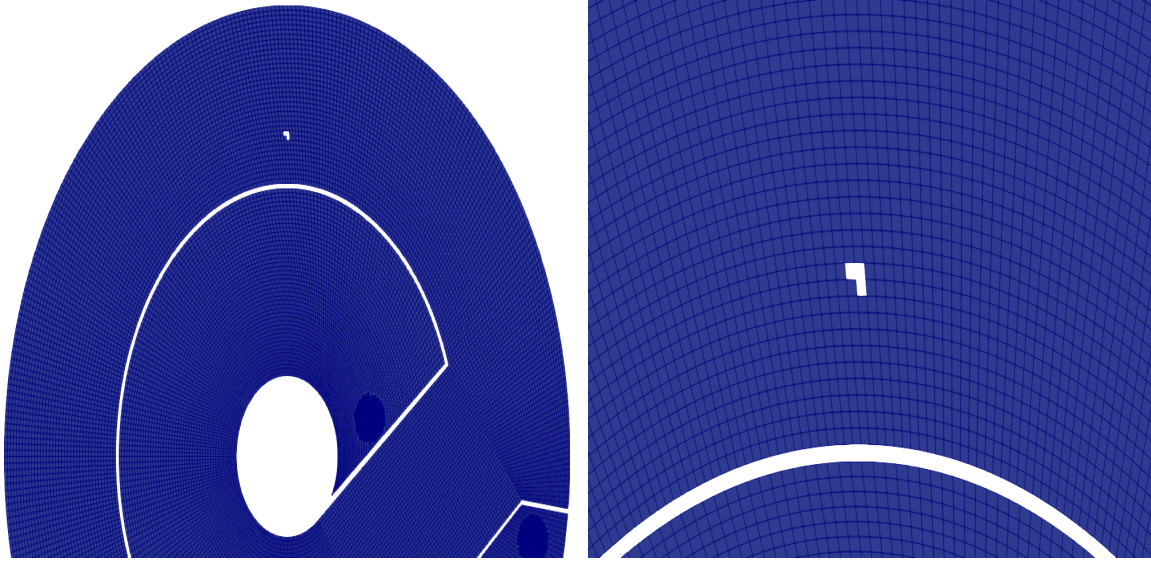


Figure 3.1.5: View of the vent seen from below the reservoir.

The location for the vent was chosen arbitrarily, and it consists of the cells (three cells) that are located within a radius of 0.2 meters. The vent was created similarly to the inlet and outlet and was defined in the *topoSetDict* file (see Appendix B.1.3 for the source code).

3.1.4 Boundary conditions

This section will explain the method of implementing the boundary conditions of the computational domain in OpenFOAM. The numerical values of the boundary conditions will differ depending on the simulation case.

Trace element analysis

A tracer was used in the incoming water to simulate the residence time of water and its movement in the reservoir. The trace element concentration was set to 1 in the incoming water and 0 in the water volume which is in the reservoir from the start. The trace element parameter is denoted s in OpenFOAM and has a unit of seconds. It is calculated with the *scalarTransport* function object which is defined in the *controlDict* file (see Appendix B.1.4 for the source code).

The trace element is modeled as a scalar with a diffusion coefficient of $1.4 \cdot 10^{-5}$, and is used to determine the distribution of the water's residence time in the reservoir. It is time-dependent and with each passing second the residence time of each water particle

in the reservoir increases with one second.

Initial water level

The initial water level (expressed as a in the code) is specified in the *blockMeshDict* since it is needed to construct the computational mesh. The initial water level is equal to the initial vertical length of the domain.

Inlet and outlet

– Velocity

The velocity boundary condition of type *flowRateInletVelocity* was used for the inlet and outlet. The time-dependent volumetric flow rates (m^3/s) of the inlet and outlet were specified in a table as a function of time. When set to 0, it implies that the inlet/outlet is closed and no water is flowing through. The boundary condition is defined in the U dictionary file located in the 0 directory (see Appendix B.2.1 for source code example).

The volumetric flow rates were calculated as

$$\dot{V} = \frac{\Delta V}{\Delta t}, \quad (3.1)$$

where ΔV is the volume increase or decrease and Δt is the time interval for the in- or outflow period. The water level changes are used to determine the volume changes which are calculated as

$$\Delta V = (h_1 - h_0) \cdot \pi r^2, \quad (3.2)$$

where h_0 is the water level before the start of the in-/outflow period and h_1 is the water level after the end of the in-/outflow period. r is the inner radius of the reservoir which is equal to 18 meters.

– Temperature

The temperature boundary conditions of the inlet (which specifies the temperature of inflowing water) are a little more complex in comparison with the velocity boundary conditions. Apart from also being time-dependent, the boundary condition is a Robin boundary condition which means that it switches between two different boundary condition types. These boundary condition types are *fixedValue* and *zeroGradient*.

The switch of boundary condition is enabled by the implementation of a coded boundary condition, the type is called *codedMixed* in OpenFOAM. A code was written to define the time intervals the *zeroGradient* boundary condition is used and the time intervals the *fixedValue* boundary condition is used. The latter takes on different values, one for each inflow period.

For the outlet, the temperature boundary condition of type *zeroGradient* is used at all times.

The temperature boundary conditions are specified in the *T* dictionary file located in the 0 directory (see Appendix B.2.2 for source code example).

– **Pressure**

The pressure boundary condition of type *fixedFluxPressure* was used for the inlet and outlet. This boundary condition sets the pressure gradient to the provided value such that the flux on the boundary is that specified by the velocity boundary condition.

The pressure boundary conditions are specified in the *p_rgh* dictionary file located in the 0 directory (see Appendix B.2.3 for source code example).

– **Trace element**

At the inlet, the trace element (*s*) boundary condition of type *fixedValue* is used. The value is set to uniform 0, which specifies that the residence time of incoming water will initially be zero.

At the outlet, the boundary condition of type *zeroGradient* is used.

The trace element boundary conditions are specified in the *s* dictionary file located in the 0 directory (see Appendix B.2.4 for source code example).

– **Turbulent properties**

At the inlet, the Dirichlet boundary condition *turbulentIntensityKineticEnergyInlet* is used for the turbulent kinetic energy *k*. Its value is set to $k = 0.003 \text{ m}^2/\text{s}^2$ which is estimated in Eq. (3.3) based on the mean flow velocity (*U*) and the turbulence intensity (*I*).

$$k = \frac{3}{2}(UI)^2 \quad (3.3)$$

The turbulence intensity is defined as

$$I \equiv \frac{u'}{U}, \quad (3.4)$$

where u' is the root-mean-square of the turbulent velocity fluctuations. The turbulence intensity can be estimated at the inlet based on the physics of the case. This case, which involves the simulation of water in a reservoir with low-speed flows (low Reynolds number), can be assumed to be a so-called *Medium-turbulence case*. For this case, the turbulence intensity is typically between 1 % and 5 % [43]. In this project, it was assumed to be equal to 3 % ($I = 0.03$).

The mean flow velocity U at the inlet is assumed to be around 1.5 m/s. With the values of U and I inserted in Eq. (3.3), the estimated value of k was calculated as $0.003 \text{ m}^2/\text{s}^2$.

The boundary condition of k is specified in the file k located in the `0` directory (see Appendix B.2.7).

Vent

The boundary conditions for the vent patch are all of type *zeroGradient* except for the pressure boundary condition which is of type *fixedValue* and set to 0.

Water level

The water level is modeled with a velocity boundary condition of type *MovingWallSlip* which allows the water level to move in the vertical direction (normal to the water level patch) while letting the tangent velocities slip (no friction along the water level).

The time-dependent vertical velocity of the water level is specified in the dictionary file *pointMotionUz* located in the `0` directory (see Appendix B.2.5 for source code example). The velocity is defined by a point motion boundary condition of type *uniformFixedValue* (allowing the value to be prescribed as a function of time). The time-varying values of the vertical velocity (U_z) are specified in a table and they were calculated as

$$U_z = \frac{\dot{V}}{A}, \quad (3.5)$$

where \dot{V} is the volumetric flow rate (through the inlet/outlet) and A is the total floor area of the reservoir which is set to 988.679 m^2 .

The temperature and trace element boundary conditions are of type *zeroGradient*. The pressure boundary condition is of type *fixedFluxPressure*.

Walls

The walls of the reservoir are modeled with the no-slip condition, meaning that the fluid will have zero velocity relative to the walls. This is applied in OpenFOAM with the velocity boundary condition of type *fixedValue* together with the value entry set to uniform (0 0 0).

The temperature and trace element boundary conditions are of type *zeroGradient*. The pressure boundary condition is of type *fixedFluxPressure*.

3.1.5 Wall functions

Wall functions are boundary conditions that are applied on the walls of the reservoir. They are used to model the flow properties in the near-wall region when using a coarse mesh. This section will describe the wall functions used in OpenFOAM.

Turbulent viscosity

The turbulent viscosity wall function called *nutkWallFunction* was used with the value set to 0. The turbulent viscosity is always zero at the walls since the turbulent kinetic energy is zero here. The wall function is specified in the file *nut* located in the *o* directory (see Appendix B.2.6).

Turbulent kinetic energy

The wall function used for the turbulent kinetic energy is called *kqRWallFunction*, it provides a simple wrapper around the zero-gradient condition.

The wall function is specified in the file *k* located in the *o* directory (see Appendix B.2.7).

Dissipation of turbulent kinetic energy

The wall function used for the dissipation of turbulent kinetic energy is called *EpsilonWallFunction*. It is used with a value of the turbulence dissipation rate set to $0.01 \text{ m}^2/\text{s}^3$. This value is also applied to the inlet and is seen as an initial guess of the incoming turbulence. The wall function is specified in the file *epsilon* located in the *o* directory (see Appendix B.2.8).

The value of $e = 0.01$ is calculated in Eq. (3.6) as

$$e = C_\mu^{3/4} \frac{k^{3/2}}{l}, \quad (3.6)$$

where the model constant C_μ is equal to 0.09, the turbulent kinetic energy k is set to $0.003 \text{ m}^2/\text{s}^2$ and l is the turbulent length scale which is set to 0.0027 meters. The length-scale definition in Eq. (3.6) is based on the mixing-length [43].

Turbulent thermal diffusivity

The wall function used for the turbulent thermal diffusivity (α_T) is called *alphatJayatillekeWallFunction* and is based on the Jayatilleke approach for determining the P quantity in the logarithmic wall function (see section 2.3.7). The value of the turbulent thermal diffusivity was set to 0 at the walls, and also at the inlet, outlet, and water level. The turbulent Prandtl number was set to 0.85. The wall function is specified in the file *alphat* located in the `0` directory (see Appendix B.2.9).

3.1.6 Solver setup

The solver which was used in OpenFOAM is called *buoyantBoussinesqPimpleFoam* and is a transient solver for buoyant, turbulent flow of incompressible fluids. The solver uses the Boussinesq approximation where the kinematic density is expressed as $\rho_k = \frac{\rho}{\rho_0} = [1 - \beta(T - T_0)]$ (see section 2.3.4). The solver uses unsteady RANS (URANS) modeling and the turbulence model used for solving the URANS equations is the $k - \epsilon$ model.

Table 3.1.2 shows the solver settings that were used in OpenFOAM. The three first settings are specified in the *fvSolution* dictionary in the *system* directory and relate to solution and algorithm control. The three last settings are specified in the *controlDict* dictionary in the *system* directory and relate to the main case control.

Table 3.1.2: Solver settings.

Setting	Option
nOuterCorrectors	1
nCorrectors	3
nNonOrthogonalCorrectors	2
Initial time step	0.01
adjustTimeStep	yes
maxCo	1

Equations are often solved multiple times within a time step. The setting `nOuterCorrectors` sets the total number of times the entire system of equations is solved within a time step (number of outer loop iterations). The setting `nCorrectors` sets the number of times the algorithm solves the pressure equation and momentum corrector in each time step (number of inner loop iterations). The setting `nNonOrthogonalCorrectors` sets the number of repeated pressure equation iterations, which is used to update the explicit non-orthogonal correction to account for mesh non-orthogonality [44].

The solver uses the PISO algorithm since one outer loop iteration implies that the PIMPLE solver will operate in PISO mode.

3.2 Mesh independence study

Simulations were initially run for a short time interval of 5-6 hours, to carry out the mesh independence study. The mesh independence study is carried out by simulating the reservoir during a complete filling and a draining, by starting with a coarse mesh and gradually refining it. To simplify these simulations, the implementation of the tracer in the model is excluded. The simulation is run on seven different mesh resolutions, where the most coarse mesh is made out of 274000 cells and the finest mesh has 4.3 million cells. Table 3.2.1 presents all the mesh resolutions that are used in the simulations of the study.

Table 3.2.1: Mesh resolutions simulated in the mesh independence study.

Simulation	Number of mesh cells
1	274,000
2	877,000
3	1,070,000
4	1,840,000
5	2,090,000
6	3,740,000
7	4,300,000

The simulation process covers a time interval of 5.586 hours and includes one three hours long filling with a volumetric inflow of $0.1 \text{ m}^3/\text{s}$ and a one-hour long draining with a volumetric outflow of $0.1 \text{ m}^3/\text{s}$.

3.2.1 Temperature averaged over outlet

Figure 3.2.1 shows the time variation of the simulated temperature averaged over the outlet, for each mesh resolution. The dotted blue line corresponds to the coarsest mesh resolution and the three solid lines correspond to the three finest mesh resolutions of 2.09 million cells, 3.74 million cells, and 4.3 million cells.

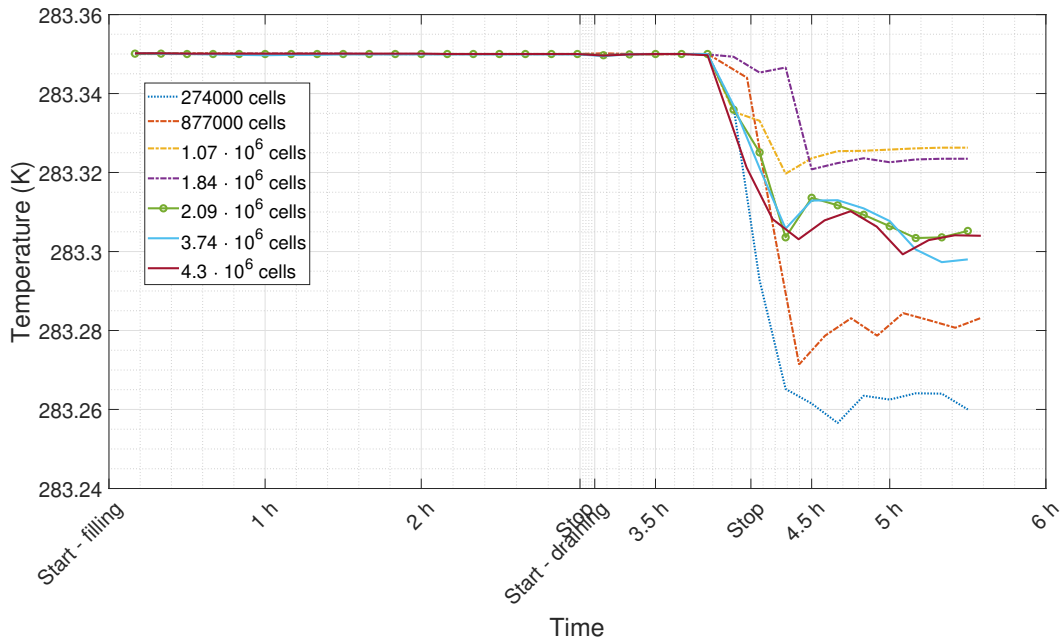


Figure 3.2.1: Average outlet temperature as a function of time, for different mesh resolutions.

It is clear that the three lines for the finest resolutions converge, and therefore one

can conclude that the simulation is mesh independent when the mesh has a minimum of around 2 million cells. The green line with circular markers corresponds to the simulation with 2.09 million cells.

3.2.2 Time-averaged temperature profile

Since the simulated thermal stratification in the reservoir is of great interest, it is also important to make sure that the simulated temperature profiles in the reservoir are mesh independent. The temperature profile at a certain point located midway between the inlet and the outlet (called mid-point) is extracted at four different times. These points in time are 10200 s (during the filling process), 14100 s (during the draining process), 17100 s (38 min after the end of the draining process), and 20100 s (1.5 hours after the end of the draining process). The average of the temperature profiles of the four different times is computed for each of the seven mesh resolutions.

Figure 3.2.2 shows the time-averaged temperature profile located at the mid-point, for each mesh resolution.

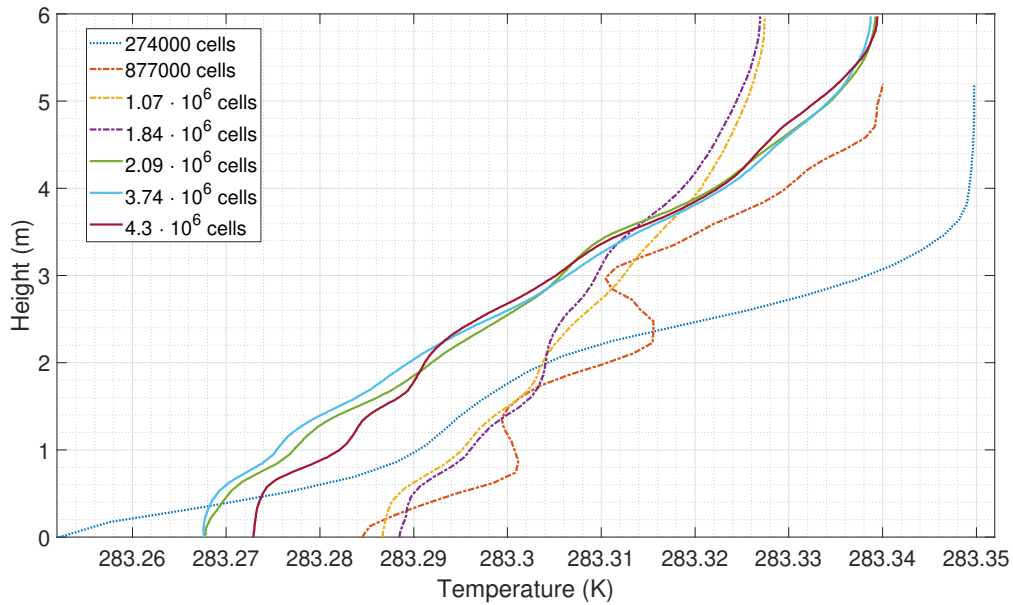


Figure 3.2.2: Time-averaged temperature profile at the mid-point for different mesh resolutions.

It can be seen that the three solid lines corresponding to the three finest resolutions more or less coincide, and the mesh consisting of 2.09 million cells is the coarsest one of the three converging simulations. This indicates that the results are mesh independent

for a mesh size of a minimum of around 2 million cells.

From the convergence plots, it is concluded that the mesh of future simulations should have around 2 million cells in order to generate mesh independent results.

3.3 Simulations of operational strategies

To examine how the flow conditions depend on different operational strategies used and also how they relate to microbial growth in the reservoir, simulations were set up according to the previous operation of the reservoir. The operating conditions (consisting of filling and draining processes) are defined by parameters such as the variation of the water level (h), the volumetric flow rate of incoming water (\dot{V}_{in}), and the temperature of incoming water (T_0) relative to the initial temperature in the reservoir (T_a). The temperature of incoming water was assumed to be dependent on temperature conditions outside the reservoir (temperature of ambient air). Flow cytometer data of previous operational conditions were extracted from Acurve (which has all the data gathered on how the reservoir has been operated by Norrvatten).

3.3.1 Simulation case setup and approach

To understand the relation with microbial growth, those periods which have demonstrated increased microbial growth (shown by the cultivation results of a water sample) were chosen to be simulated. Since some initial conditions were guessed when setting up the simulation, it was decided to simulate a longer period to get more accurate results. So instead of just simulating an interesting filling process that occurs before a water sample is taken, the simulation was run a few days before this specific point of time when a water sample was taken.

Table 3.3.1 shows the periods chosen for simulation. Desired simulation period was seven days for all the simulations meaning that all the simulations were set up to run seven days according to input data from Acurve. However, a majority of the simulations failed to run the full run time due to mesh instability. Simulations tended to crash due to a too high Courant number. In Table 3.3.1, *End time* specifies at what time step the simulation ended. See section 5.2 for a more detailed failure analysis of the simulations. During each period, a water sample was taken. The results of the water sample are either assessed as "potable" or "potable with reservations" depending

on if the microbial growth exceeded the limit values determined by the regulations of the Swedish Food Agency.

Table 3.3.1: Simulation and water sample details of the chosen seven-day periods.

Desired simulation period	Mesh size (No. of cells)	End time (Days HH:MM:SS)	Time of water sampling	Water sample result
12-18 June 2021	2,013,917	1 15:36:10	2021-06-16 11:35	Potable
16-22 September 2021	172,518	5 03:11:40	2021-09-20 12:00	Potable with reservations
22-28 September 2021	2,096,987	7 00:00:00	2021-09-28 10:45	Potable with reservations
7-13 October 2021	2,137,709	3 18:05:50	2021-10-13 11:10	Potable with reservations

The simulation of the period 16-22 September 2021 has a smaller mesh size (172,518 cells) than the other simulations since this simulation was run before the mesh independence study. Therefore, the results of this particular simulation may not be as accurate since they are not mesh-independent.

Water samples

Table 3.3.2 shows the cultivation results of each of the water samples specified in Table 3.3.1. The cultivation of the water samples has been performed by Norrvatten.

Table 3.3.2: Cultivation results of water samples.

Time of sampling	Sample number	Sample ID	Sampling temperature (°C)	Culturable microorganisms 22 °C, 3 days	Slow-growing bacteria 22 °C, 7 days
2021-06-16 11:35	56895	NV.PR.12-PP_433-210616-3	9	1	10
2021-09-20 12:00	59319	NV.PR.12-PP_433-210920-2	-	2	8700
2021-09-28 10:45	59526	TORN-PP_433-210928-1	10	3	18000
2021-10-13 11:10	60014	NV.PR.12-PP_433-211013-1	10.9	180	560

As seen in the table, the period 12-18 June 2021 resulted in a water sample that demonstrated an approved level of microbial growth. It was decided to simulate this period to be able to compare the operational strategies of a 'successful week' with the operational strategies of a less successful week. However, the main focus was on simulating periods which have demonstrated increased microbial growth to recognize problematic operational strategies of the reservoir.

Analysis of mixing processes with density effects

Each of the seven-day periods defined in Table (3.3.1) has several filling and draining processes. The main focus was on examining the filling processes (the inflow parameters and conditions) to investigate the mixing processes. A mixing process is defined as the phase when incoming water mixes with the reservoir water due to turbulence being created by the turbulent buoyant jet that is the inflowing water. In this study, a mixing process is defined as the period between the start of a filling process to the start of a draining process. It can behave differently and last for various intervals of time, depending on the inflow conditions. Inflow conditions are defined by the inflow velocity (U_0), the density of inflowing water (ρ_0), inlet diameter (d), the density of reservoir water (ρ_a) and initial water depth ($h_w(0)$) in the reservoir. The mixing processes are investigated to see if there are some stagnant zones in the reservoir and to analyze if the inflow conditions can be changed in a specific way to avoid stagnation. The potential risk of stagnation in the reservoir is water quality degradation caused by excessive aging, which is due to insufficient water exchange in the reservoir.

In total, six filling processes of each desired simulation period were simulated or supposed to be simulated. Due to different circumstances, not all filling processes simulated are of interest in the analysis. Simulations that were run with a smaller mesh size than the mesh-independent size of 2 million cells were disregarded in this analysis. Therefore, simulations from the period 16-22 September 2021 were not included. Filling processes that were run at the beginning of the simulation period (during the two first days) were also disregarded due to less accuracy. Figure 3.3.1 shows the filling processes which are of interest in the analysis. They belong to the desired simulation periods of 22-28 September and 7-13 October 2021.

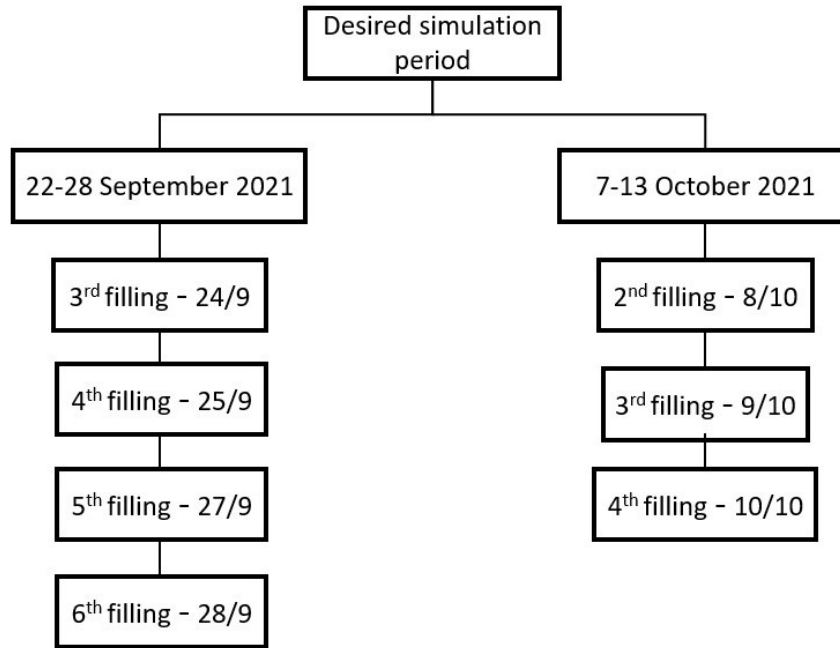


Figure 3.3.1: Filling processes used in the analysis of mixing processes.

As seen in the figure, only three filling process simulations of two of the simulation periods were chosen to be analyzed. Both of the simulation periods included a water sample result which was assessed as "potable with reservations". The simulation of the period 12-18 June 2021 could not be analyzed since the simulation crashed after 142,570 seconds (1.65 days) which is during the first draining process. Since only the first filling process managed to be simulated, there are no accurate simulation results of this period.

3.3.2 Initial conditions

Simulations of the filling processes specified above (and the desired simulation periods) were set up according to the operational strategies that were used by Norrvatten when operating the reservoir during those periods. The information on the used operational strategies is registered in the reservoir by the BactoSense flow cytometer. That data were then extracted from Acurve and used as initial boundary conditions when setting up the simulations.

The plots below in Figure 3.3.2 and 3.3.3 show the water level variation in the reservoir during each of the filling processes. The data have been extracted from Acurve. Each of the filling processes is followed by a draining process.

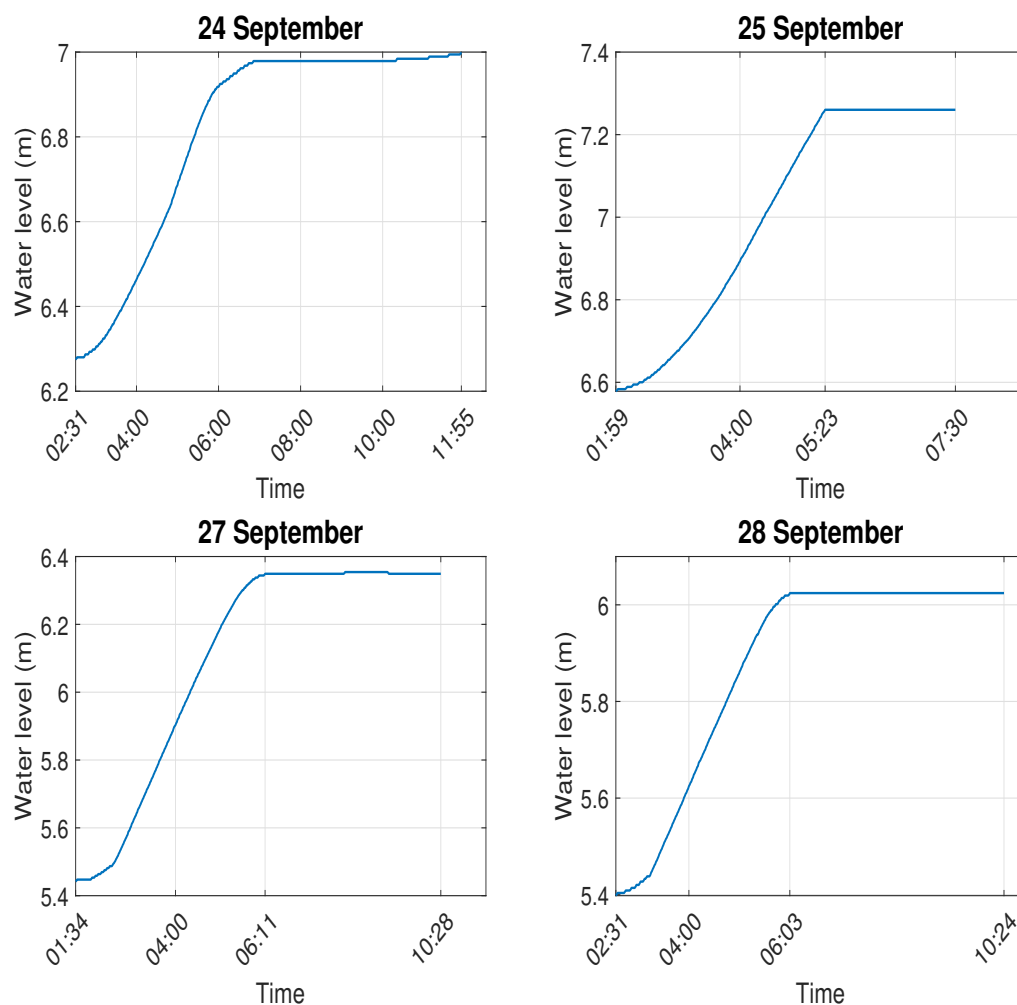


Figure 3.3.2: Water level variation during four succeeding filling processes of the period 22-28 September 2021.

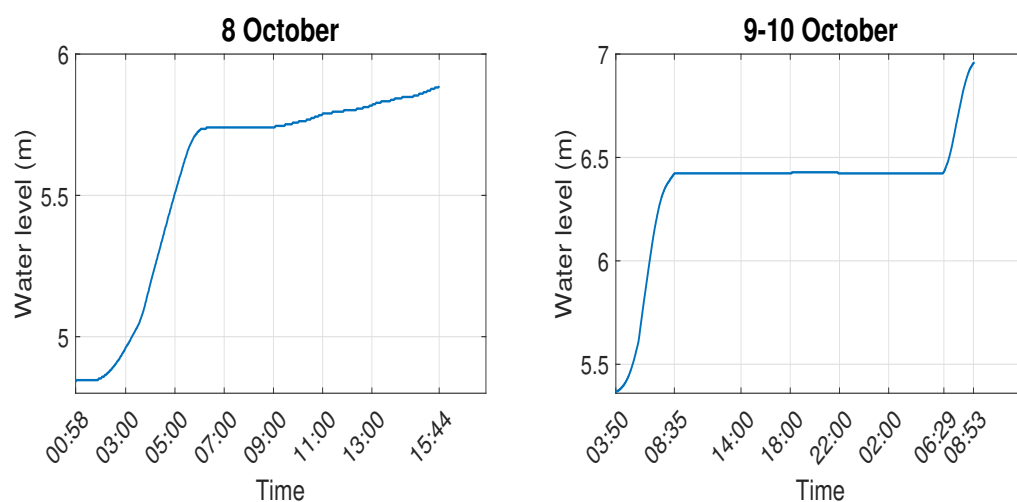


Figure 3.3.3: Water level variation during three succeeding filling processes of the period 7-13 October 2021.

The right plot in Figure 3.3.3 differs from the rest in the sense that it shows two filling processes, one started the 9th October and the other one started the 10th October. Since the first filling process is followed by another filling process rather than a draining process, these two are put together in their analysis. The focus is on analyzing the mixing process before the water is drained.

Table 3.3.3 shows the remaining required boundary conditions and inflow conditions of each filling process. The data are either directly extracted from Acurve, calculated, or estimated according to section 3.1.4.

Table 3.3.3: Inflow parameters of each filling process.

	Initial water depth (m)	Inflow rate (m ³ /s)	Inflow velocity (m/s)	Water level velocity (m/s)	Temperature of inflowing water (K)	Initial temperature (K)
3 rd filling 24/9	6.275	0.0218	0.0771	2.2042e-05	283.067	283.209
4 th filling 25/9	6.578	0.0567	0.2005	5.7348e-05	282.816	283.195
5 th filling 27/9	5.442	0.0556	0.1966	5.6191e-05	283.472	283.169
6 th filling 28/9	5.398	0.0501	0.1772	5.0675e-05	283.726	283.220
2 nd filling 8/10	4.841	0.02	0.0707	2.0252e-05	284.444	284.449
3 rd filling 9/10	5.364	0.0631	0.2232	6.3777e-05	284.143	284.448
4 th filling 10/10	6.430	0.0622	0.2200	6.2868e-05	284.394	284.382

The water level velocity in Table 3.3.3 is the vertical velocity of the water level which is calculated as in Eq. (3.5). As water is either entering or leaving the reservoir, the water level will have a vertical velocity that is consistent with the in- or outflow of water. The velocity is unique for each filling/draining process and is constant during an entire filling/draining process.

The temperature data marked in blue indicate associated filling processes that have a lower temperature of inflowing water relative to the initial temperature of water in the reservoir ($T_0 < T_a$). Colder inflow (having a higher density) may result in stagnation due to a stable thermal stratification in the reservoir.

Another factor that influences the inflow conditions is the design of the inlet. However, the diameter of the inlet was kept constant in all the simulations and was set to 0.6 meters. The inlet is vertical, giving the better condition for the mixing process compared to a horizontal inlet.

3.4 Simulation with alternative operational strategies

To understand in which way the operational strategies can be adjusted to increase the exchange of water and prevent microbial growth, simulations with alternative operational strategies were set up. The inflow period during 28th September was chosen to be run with changed operational strategies. The reason for choosing the 28th September is that a water sample was taken by Norrvatten this day (after the inflow period) which showed that microbial growth of slow-growing bacteria exceeded the limit values set by the Swedish Food Agency's regulations on drinking water. Therefore, it was desirable to investigate how the flow conditions of this specific inflow period can be improved to avoid getting a water sample result showing increased microbial growth.

The inflow parameters that were changed when setting up the alternative simulations were:

- The volumetric inflow rate (\dot{V}_{in})
- Water level variation (h)
- Temperature of inflowing water (T_0)

3.4.1 Simulation case setup

Four alternative setups of the inflow period were created. The original simulation and the four alternative setups are defined in Table (3.4.1) based on their inflow conditions.

Table 3.4.1: Inflow conditions of the original and alternative simulation setups of the inflow period the 28th September.

Simulation	t_{start} (hh : mm)	t_{end} (hh : mm)	T_0 (K)	\dot{V}_{in} (m ³ /s)	h_o (m)	h_i (m)
Original	02:31	06:03	283.7261	0.05	5.398	6.024
Strategy 1	02:31	03:41	283.7261	0.15	5.398	6.024
Strategy 2	02:31	06:03	283.7261	0.15	5.398	7.276
Strategy 3	02:31	06:03	283.0	0.05	5.398	6.024
Strategy 4	00:28	04:00	283.7261	0.05	5.398	6.024

Table values which are marked in red correspond to changed inflow conditions, which

have been changed according to the alternative operational strategies. t_{start} and t_{end} stand for the start time and end time of the inflow period. T_0 is the temperature of the inflowing water, \dot{V}_{in} is the volumetric inflow rate, h_0 is the initial water level and h_1 is the water level after the inflow period.

- **Strategy 1**

The first approach to changing the operational strategies was to increase the volumetric inflow rate (\dot{V}_{in}) to generate higher flow speeds and stronger turbulence. A more turbulent jet flow will work as a more efficient mixer. The volumetric inflow was increased by three times (from 0.05 to 0.15 m³/s). The water level was kept at the same values as in the original simulation, resulting in a three times shorter inflow period.

- **Strategy 2**

In the Strategy 2 simulation, the inflow rate was also increased to 0.15 m³/s. However, now the time interval of the inflow period was kept the same as in the original simulation (with the same end time). This resulted in three times larger volume flowing into the reservoir during the whole inflow period, which led to a three times larger water level increase (resulting in a water level of 7.276 m after the inflow period).

- **Strategy 3**

The third approach was to study the effect of a different temperature condition by setting the temperature of the inflowing water to another value. In the original simulation, the inflowing water was warmer than the reservoir water (see Table 3.3.3). To see how a colder inflow can cause deteriorated flow conditions and eventually lead to a decreased exchange of water, the Strategy 3 simulation was chosen to be simulated with a lower temperature of the inflowing water. The temperature of the inflowing water was changed to 283.0 K (0.22 K colder than the reservoir water which has a temperature of 283.22 K).

- **Strategy 4**

The fourth and last approach was to decrease the water level before the time the water sample was taken by Norrvatten (at 10:45) by moving the following outflow period forward in time. To do that, the inflow period also needed to be moved forward (with

a start time at 00.28 and an end time at 04:00). The outflow rate was also increased by a factor of 30 (from -0.001 to $-0.03 \text{ m}^3/\text{s}$), while the time interval of the outflow period was decreased by $2/3$. This resulted in a ten times bigger water level decrease after the outflow period (which ends at 09:15). The changes resulted in a water level of 5.662 m at the time the sample was taken (instead of a water level of 6.023 m).

3.5 Post-processing

This section will explain the post-processing procedure for all of the simulations run. When the simulations were finished, each filling process was post-processed in ParaView. Focus was on visualizing the temperature and residence time distribution in the reservoir, mainly in the vertical direction since the density effects are existing in this direction. Vertical slices were created at three locations: at the inlet, midway between the inlet and outlet, and at the outlet (see Figure 3.5.1).

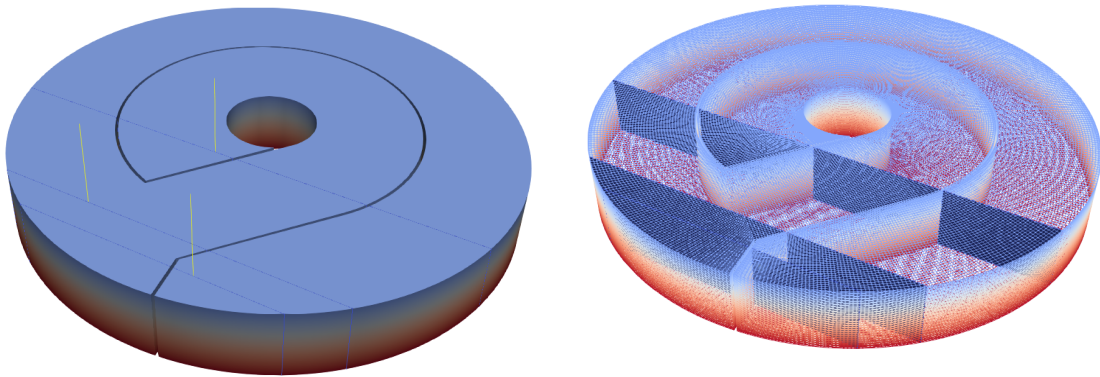


Figure 3.5.1: Reservoir model with each vertical slice at the inlet, mid-point, and outlet marked.

The three yellow vertical lines correspond to the lines over which the simulation data were plotted, one for each vertical section. Two of the lines coincide with the inlet and outlet, respectively. The location for the three lines will henceforth be referred to as the inlet, mid-point, and outlet.

To study the mixing conditions of a filling process, the flow conditions were analyzed at various times during the inflow period, with a focus on analyzing the last time step before the start of the outflow period.

Simulation data from the whole simulations as well as from the vertical slices were

extracted and post-processed in MATLAB. The simulation results of interest when post-processing is:

- Residence time s
- Temperature T
- Velocity U

3.5.1 Residence time of outgoing water

To estimate what the average residence time of outgoing water is going to be, the average residence time over the outlet is computed, both during inflow and outflow.

The maximum residence time in the reservoir is of interest in the study of the residence time distribution and especially when assessing the average residence time of water located at the outlet. By studying how much of the 'oldest' water has reached the outlet, it can be estimated how effective the exchange of water is in the reservoir. A good exchange of water in the reservoir can be characterized by having the highest residence time distributed close to the outlet.

3.5.2 Vertical distributions

To better understand the vertical temperature and residence time distribution throughout the whole reservoir, a few more graphs were plotted in MATLAB. The graphs were produced with the purpose to show:

- The vertical distribution of the residence time at different locations.
- The vertical distribution of the temperature at different locations.

By extracting simulation results from ParaView of the three vertical slices through the inlet, mid-point, and outlet (for all the time steps) the time-dependent distributions of the quantities of interest could be analyzed. Temperature and residence time (vertical) distributions were plotted against the z -axis (height) for certain times of interest. This was done for all three sections to get an overview of the flow conditions. However, it was later decided that only simulation data at the outlet and from the last time step of each mixing process was of interest when evaluating the flow conditions. This was also decided to minimize the simulation data being analyzed.

3.5.3 Comparison with flow cytometer data

Apart from the post-processing in ParaView and MATLAB of the simulation parameters of interest, the parameters are also compared to flow cytometer data that have been gathered by Norrvatten. The comparison with flow cytometer data is only relevant when analyzing the seven-day simulations that were set up in accordance with previous operating cases. The quantities measured by the flow cytometer BactoSense which are of interest are:

- HNAP (Percentage of HNA bacteria)
- HNAC (Content of HNA bacteria)
- TCC (Total cell count)

These parameters are all related to the microbial growth in the reservoir, and their role in assessing the water quality has been explained in section 2.1.5. The aim was to see if there is any correlation between microbial growth and the exchange of water in the reservoir. More specifically, if there is any connection between an increase in HNAP and an increase in residence time of the reservoir water.

Chapter 4

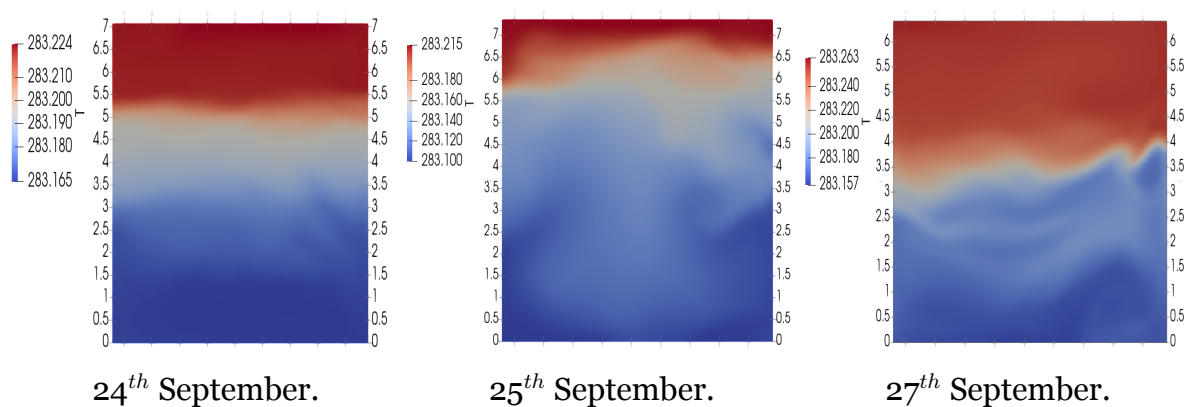
Results

4.1 Simulations of operational strategies

In this section, results from the simulations that were set up according to previous reservoir operation used in fall 2021 (methodology explained in section 3.3.1) will be presented.

4.1.1 Temperature distribution

Figure 4.1.1 shows the temperature distribution at the outlet at the last time step of each of the mixing processes.



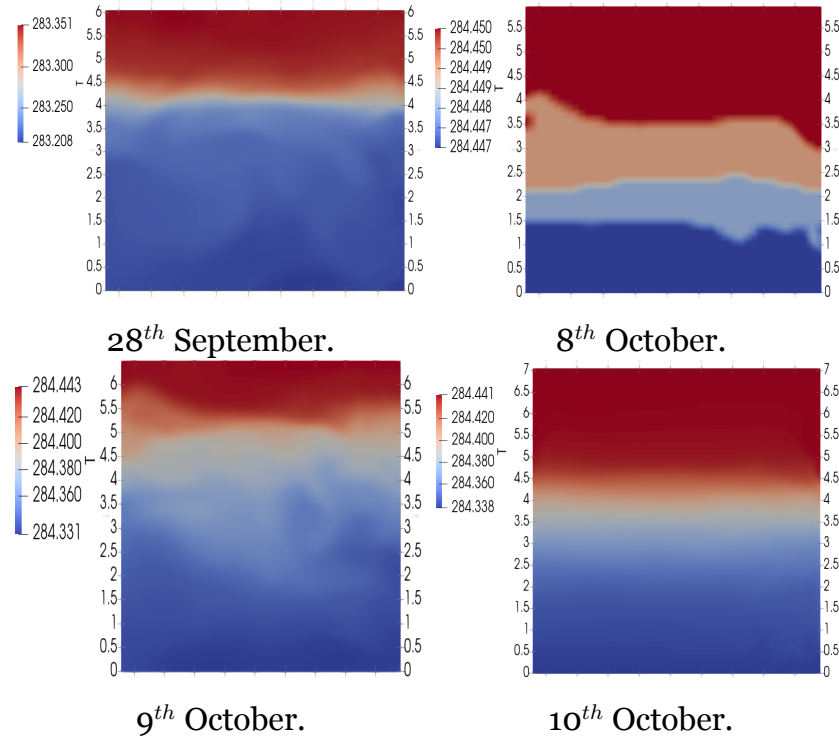


Figure 4.1.1: Simulated temperature distribution at the outlet at the last time step of each of the mixing processes.

By studying the temperature distribution in the last time step of each mixing process (just before the start of the draining process), it can be seen that a higher difference between the temperature of incoming water and the initial temperature in the reservoir will result in a larger temperature variation (ΔT) in the vertical direction at the outlet at the end of the mixing process. Also, it can be seen that a longer mixing process (longer time interval between the start of filling and the start of draining) results in a smaller temperature variation in the vertical direction at the outlet. This is due to temperature equalization.

See Appendix (C.1) for all of the time steps during each mixing process.

4.1.2 Residence time distribution

In Figure 4.1.2 and 4.1.3 the residence time distribution in a vertical slice through the outlet can be analyzed by studying the adjacent scale which shows the current residence time of the water at a particular time. The residence time is computed in seconds from the start of the desired simulation period (as defined in Figure 3.3.1). The scale ranges from the lowest local residence time (dark blue) to the highest local residence time (red).

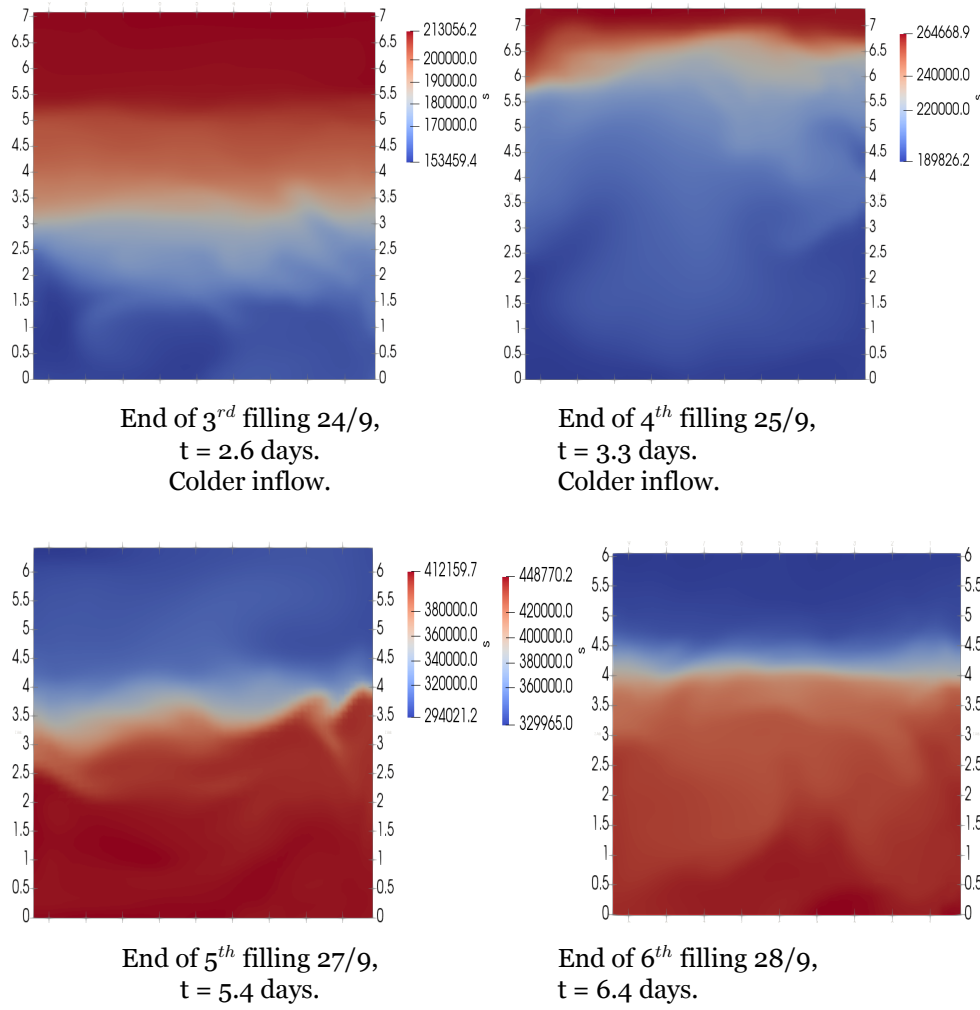


Figure 4.1.2: Simulated residence time distribution in a vertical section through the outlet at the end of the mixing process for each of the four filling processes of simulation period 22-28 September 2021.

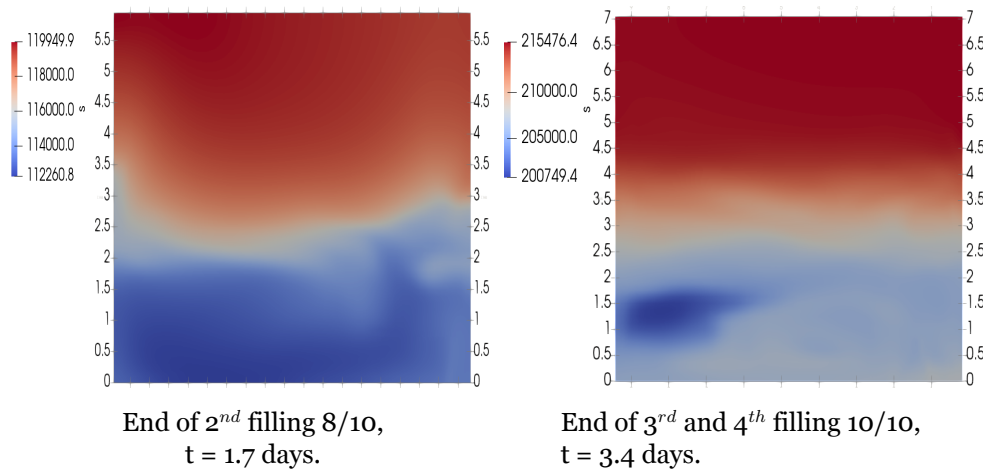


Figure 4.1.3: Simulated residence time distribution in a vertical section through the outlet at the end of the mixing process for each of the three filling processes of simulation period 7-13 October 2021.

By studying the residence time distribution in a vertical section through the outlet, it can be seen that the mixing processes which have an inflow of colder water will at the end of the mixing process have a 'problematic' residence time distribution at the outlet as the 'newer' water will settle to the bottom of the reservoir (see Figure 4.1.2). 'Newer' water is identified as having a lower residence time and is therefore colored in a darker blue color.

Larger temperature differences between incoming water and reservoir water ($T_0 - T_a$) will result in a larger residence time variation (Δs) in the vertical section through the outlet. However, a longer mixing process (larger mixing time Δt) will naturally result in a smaller residence time variation (and temperature variation) due to temperature equalization.

In Table (4.1.1) the data from the temperature and residence time distributions at the outlet are presented. The residence time difference in a vertical section is defined as $\Delta s = s_{max} - s_{min}$, where s_{max} is the maximum residence time and s_{min} is the minimum residence time of the water at a certain time. The temperature variation is defined similarly as $\Delta T = T_{max} - T_{min}$.

Table 4.1.1: Simulated residence time and temperature data in a vertical section through the outlet, at the last time step of each inflow period.

Inflow period	Δt (h)	$T_0 - T_a$ (K)	s_{min} (s)	s_{max} (s)	Δs (s)	ΔT (K)
3 rd filling 24/9	12.6	-0.142	153,459	213,056	59,597	0.059
4 th filling 25/9	5.4	-0.379	189,826	264,669	74,843	0.115
5 th filling 27/9	8.8	0.303	294,021	412,160	118,139	0.106
6 th filling 28/9	7.8	0.506	329,965	448,770	118,805	0.143
2 nd filling 8/10	16.7	-0.005	112,261	119,950	7,689	0.003
3 rd filling 9/10	26.5	-0.305	195,513	206,506	10,993	0.113
4 th filling 10/10	2.5	0.012	200,749	215,476	14,727	0.103

The maximal difference in residence time is $\Delta s = 118,800$ seconds, which equals 1.375 days. This is at a time step during the 6th filling the 28th September which corresponds to a simulated time of 6.43 days. A difference of 1.375 days in residence time between the surface and bottom of the reservoir is quite small relative to the

maximum residence time in the reservoir which is 6.43 days. This corresponds to the residence time difference being 21 % of the maximum residence time in the reservoir.

4.1.3 Profiles at outlet

Figure 4.1.4 and 4.1.5 show the temperature and residence time profiles at the outlet at the last time step of each mixing process.

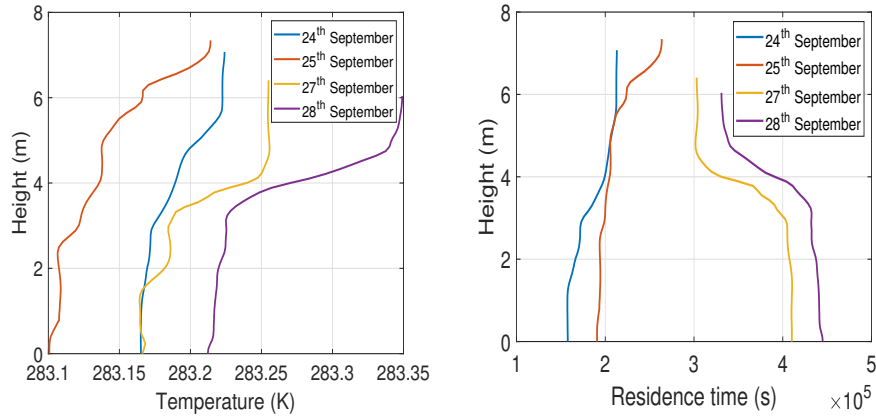


Figure 4.1.4: Vertical distribution of temperature (left) and residence time (right) at the last time step of each of the mixing processes within the period 22-28 September.

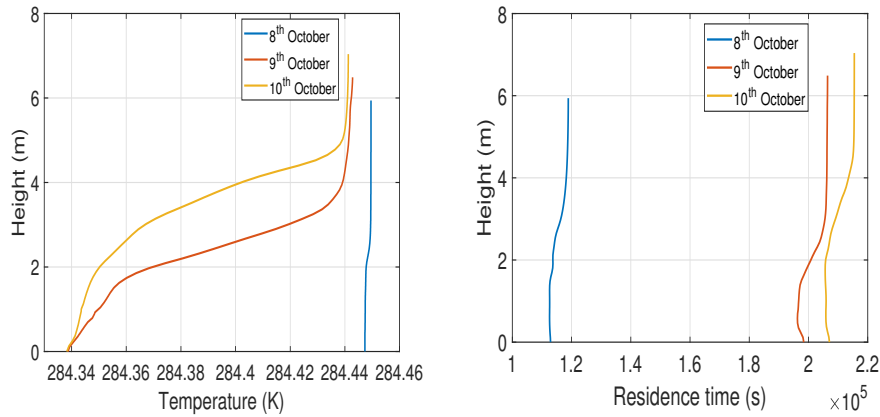


Figure 4.1.5: Vertical distribution of temperature (left) and residence time (right) at the last time step of each of the mixing processes within the period 7-13 October.

In the right plot in Figure 4.1.4 it can be seen that the mixing processes during the 27th and 28th September show a better residence time distribution with older water at the bottom of the reservoir close to the outlet. This also proves that the flow conditions followed by an inflow of warmer water are more favorable than those followed by an inflow of colder water (24th and 25th September). It can also be observed that during the mixing process of 8th October the temperature and residence time difference is small. This may be explained by the long mixing process (> 16 h).

4.1.4 Residence time at outlet

In Figure 4.1.7 and 4.1.8 the averaged residence time at the outlet is plotted together with the maximum residence time in the reservoir (as a function of time) for each of the filling processes. The last time step corresponds to the time just before the outflow period starts. By extracting the residence time at the outlet during a filling process one can get an indication of how incoming water is distributed in the reservoir from the start of the filling process until the start of the draining process.

22-28 September 2021

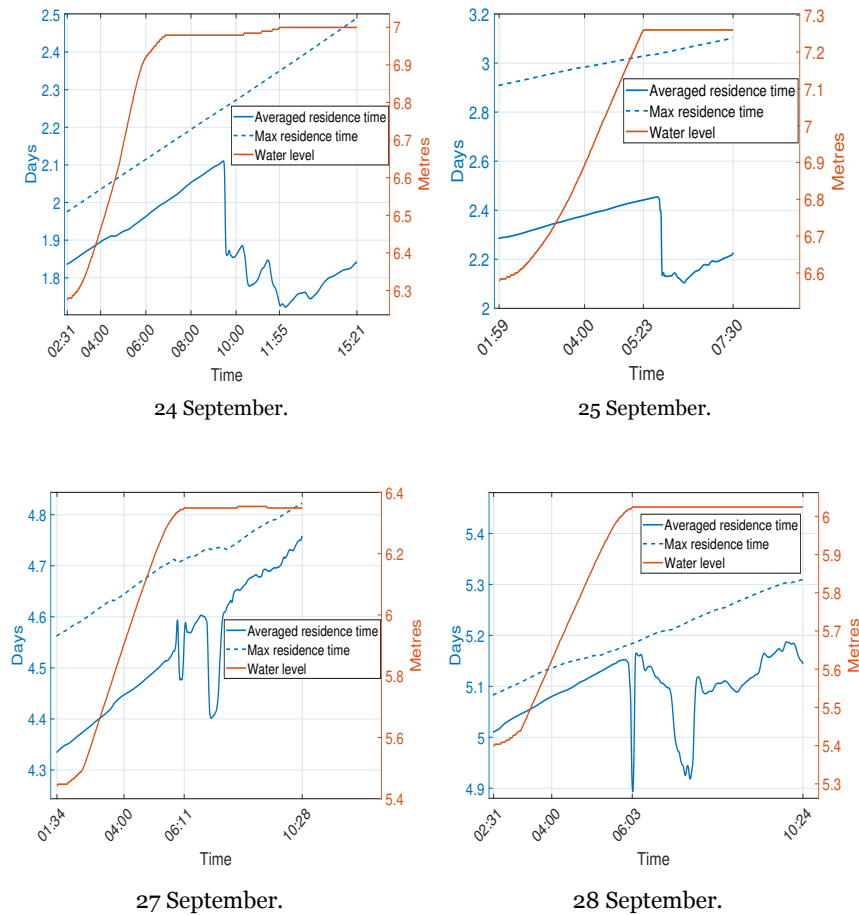


Figure 4.1.7: Simulated average residence time (age) at outlet and maximum residence time in the reservoir plotted together with the water level variation for each of the four inflow periods of simulation period 22-28 September 2021.

7-13 October 2021

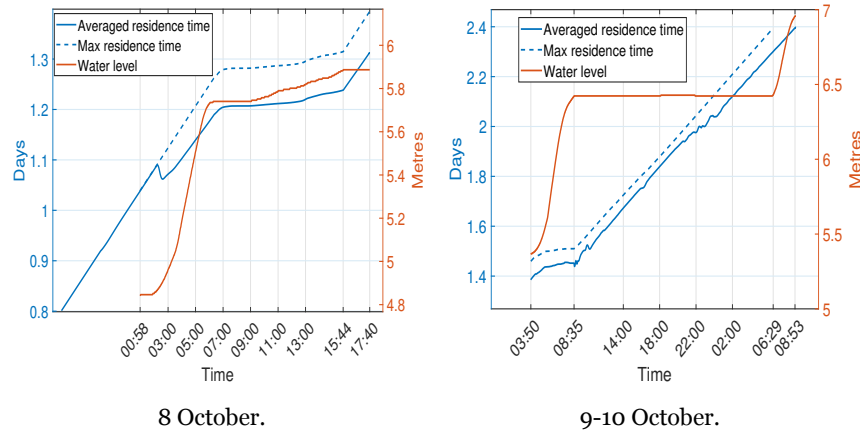


Figure 4.1.8: Simulated average residence time (age) at outlet and maximum residence time in the reservoir plotted together with the water level variation for each of the three inflow periods of simulation period 7-13 October 2021.

It can be seen that for the filling processes with a colder inflow (24th and 25th September), the average residence time at the outlet decreases towards the end of the filling process. This can be explained as an effect of thermal stratification, over time the colder 'newer' water sinks to the bottom of the reservoir due to its higher density.

By comparing the end value of the maximum residence time s_{max} with the end value of the averaged residence time at outlet \bar{s}_{outlet} , the exchange of water during a filling process (up until the start of the draining process) can be estimated. A small difference between the values ($\Delta s = s_{max}(end) - \bar{s}_{outlet}(end)$) indicates that the exchange of water is adequate. See Table 4.1.2 for values of Δs of the different filling processes.

Table 4.1.2: Difference between the maximum residence time and the averaged residence time at the outlet, at the last time step of a filling process.

Filling process	Δs (days)
24 September	0.66
25 September	0.88
27 September	0.07
28 September	0.16
8 October	0.08
9-10 October	0

Understandably, the filling processes with colder inflow (24th and 25th September) has

a much lower exchange of water (larger Δs).

Residence time of outgoing water

One way to examine the residence time of water leaving the reservoir is to extract the average residence time at the outlet during the draining process (unlike during the filling process).

Figure 4.1.10 and 4.1.11 shows the average residence time at the outlet during the draining processes of the two simulated periods (24-28 September and 8-10 October). Each draining process follows a filling process.

22-28 September 2021

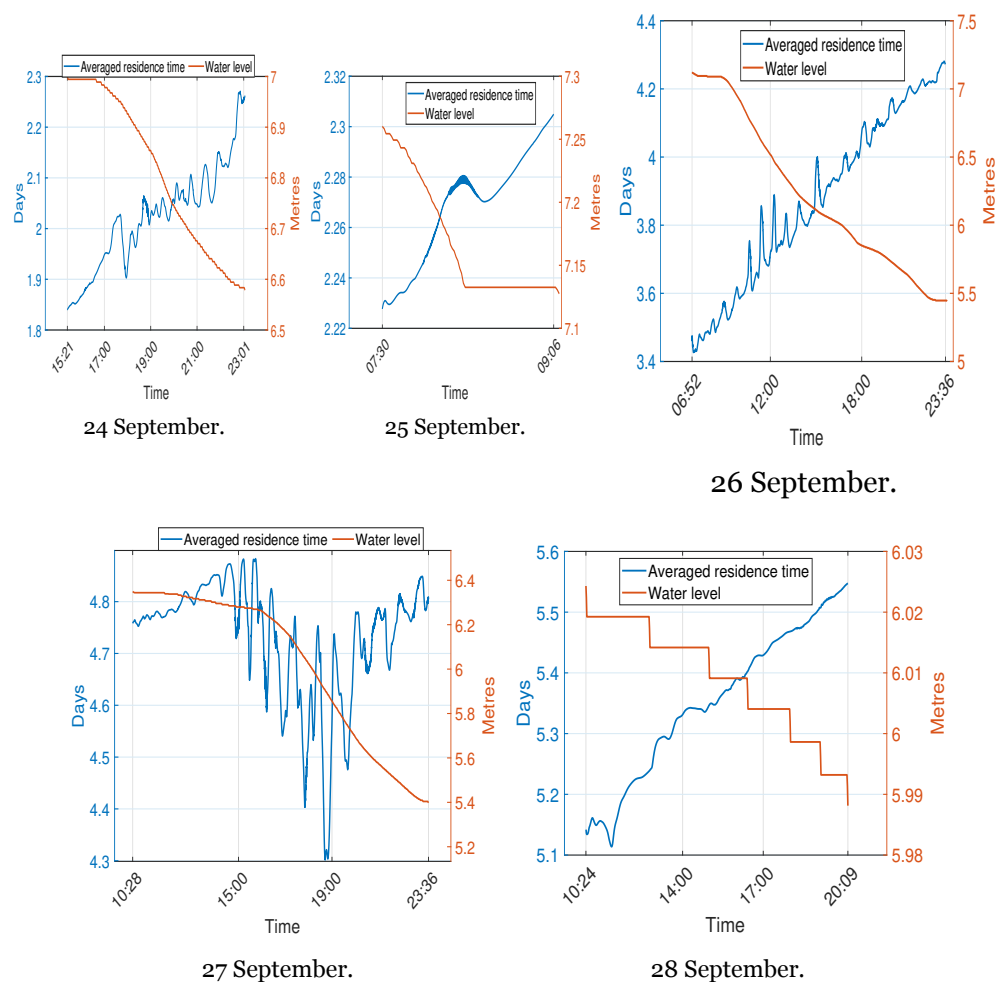


Figure 4.1.10: Simulated average residence time (age) at outlet during the draining processes of simulation period 22-28 September 2021.

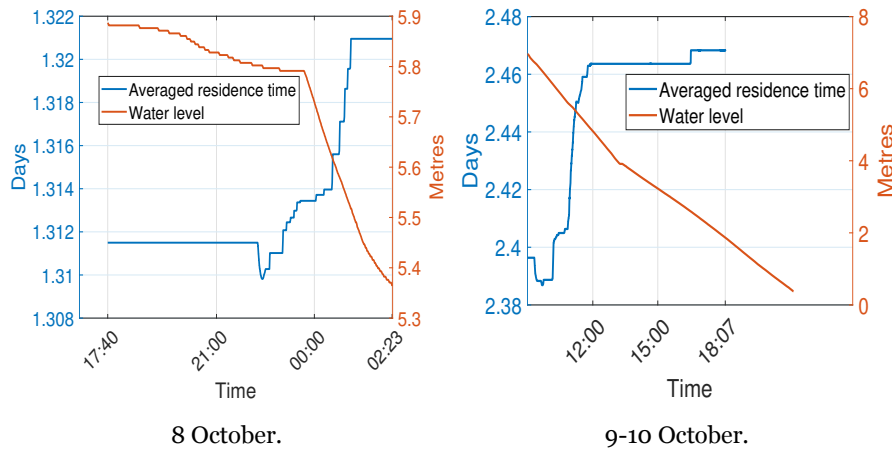
7-13 October 2021

Figure 4.1.11: Simulated average residence time (age) at outlet during the draining processes of simulation period 7-13 October 2021.

In most cases, the residence time of outgoing water is increasing during the draining process and reaches its maximum value at the end of the draining process. For these cases, the newest water is located at the bottom of the reservoir just before the outflow starts (meaning that inflowing water is colder than the initial reservoir water). However, the draining process during the 27th September (which occurs after the 5th filling process) demonstrates a different residence time variation of the outgoing water. The outgoing water which leaves the reservoir first (located at the bottom of the reservoir) is older relative to the remaining outgoing water (which is emptied later). This can be explained by the fact that the incoming water during the 5th filling process is distributed differently in the reservoir as the incoming water is warmer than the initial reservoir water. The initial reservoir water which consists of water from the colder inflow filling processes the 24th and 25th September (3rd and 4th filling) is then located at the bottom of the reservoir after the filling process.

4.2 Alternative operational strategies

In this section, the results of the simulations with alternative operational strategies will be presented. The filling process on the 28th September was chosen to be run with changed operational strategies.

4.2.1 Temperature distribution

To visualize the mixing process and how it differs between the four different operational strategies, the temperature distribution at the outlet is extracted at various times during the mixing process of each strategy (see Appendix C.1.1).

See Figure 4.2.1 for the outlet temperature distribution at six different time steps during the mixing process of Strategy 1.

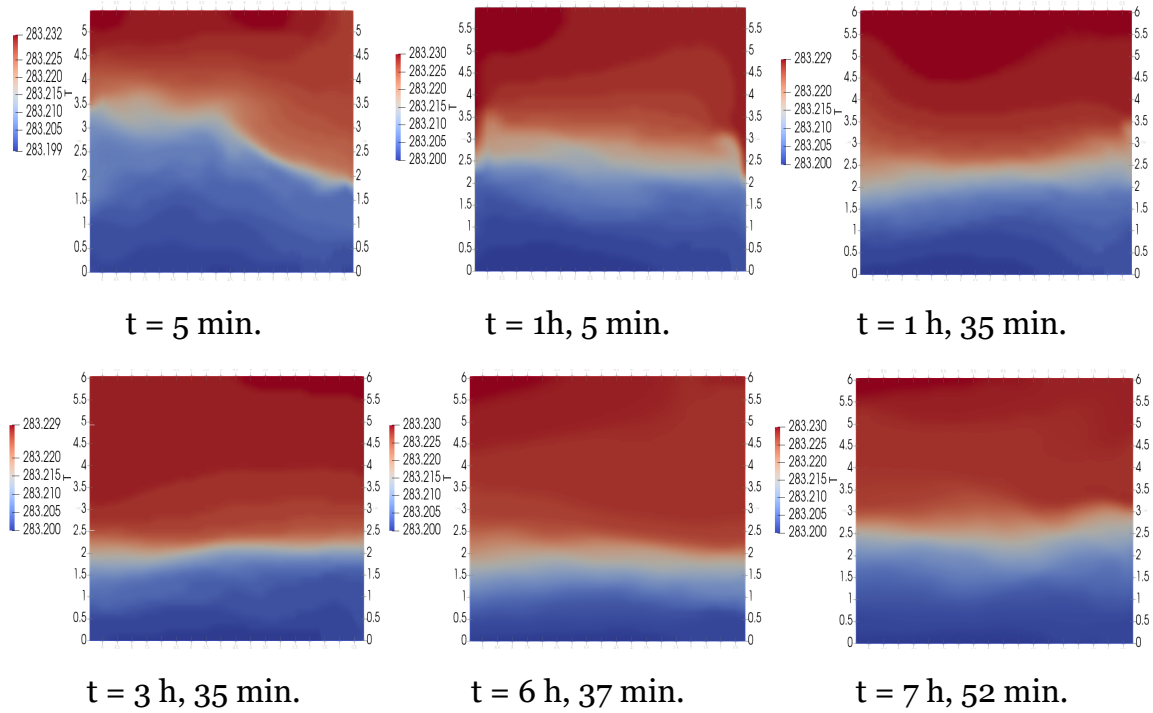


Figure 4.2.1: Simulated temperature distribution at the outlet for the Strategy 1 mixing process.

Strategy 1 differs from the rest by having the smallest vertical temperature variation ($\Delta T = 0.03 \text{ K}$) at the outlet at the last time step of the mixing process.

4.2.2 Residence time distribution

Figure 4.2.2 shows the residence time distribution at a vertical section through the outlet (at the last time step of the filling process) for the four cases that were simulated with different operational strategies.

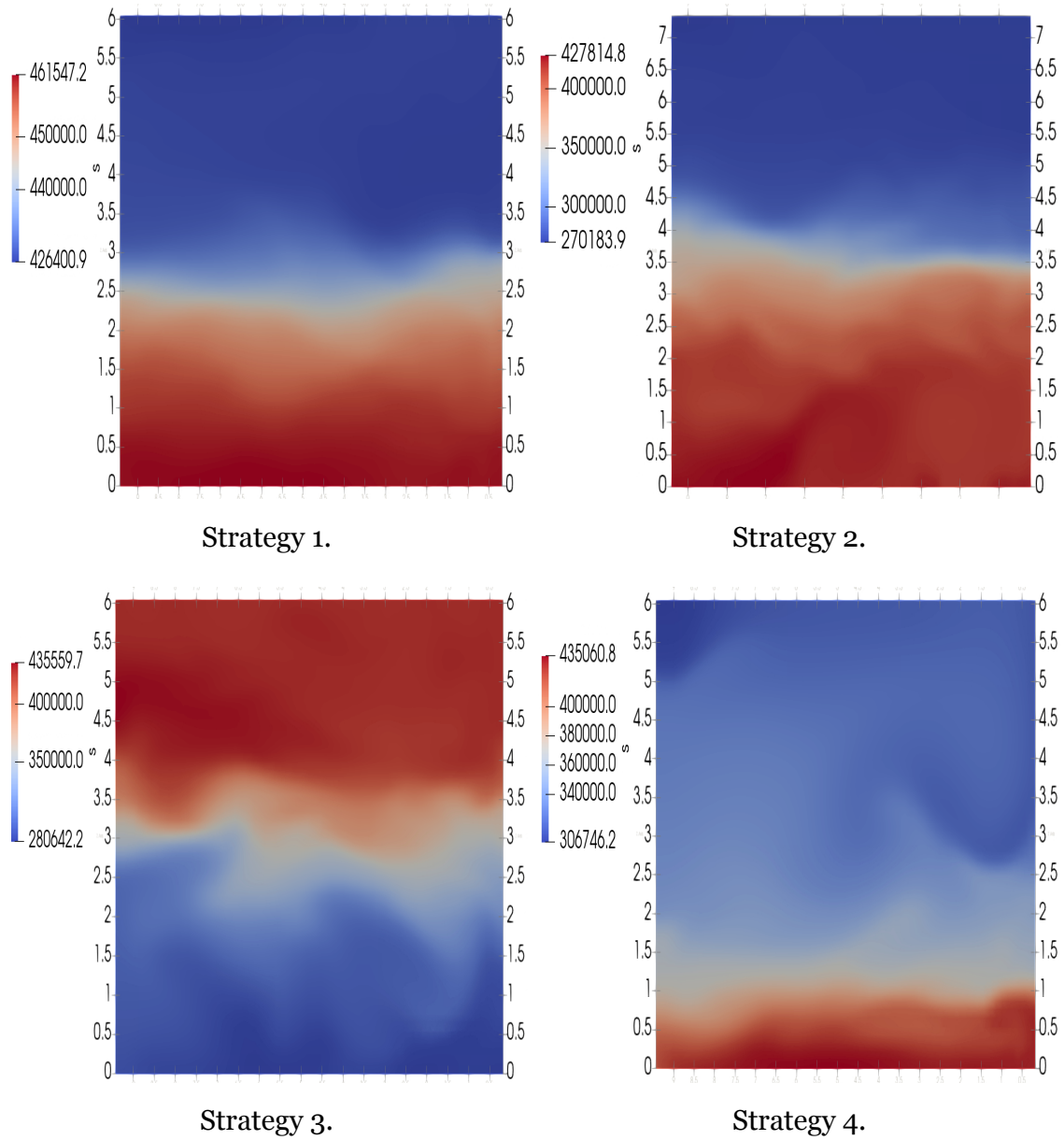


Figure 4.2.2: Simulated residence time distribution in a vertical section through the outlet at the end of the mixing process for each of the four cases with different strategies.

The original simulation of the filling process on the 28th September 2021 had a maximum residence time s_{max} equal to 448700 seconds and a minimum residence time s_{min} equal to 329965 seconds at the end of the mixing process (see Figure 4.1.2). By comparing those numbers with the residence time distributions of the alternative simulations (in Figure 4.2.2), it can be seen that the Strategy 1 simulation is the only one which has 'older' water at the bottom of the reservoir, compared to the original simulation. s_{max} is larger for the Strategy 1 simulation (461547 seconds) than for the original simulation (448700 seconds). This indicates that the operational

strategies defined by the Strategy 1 case provide a better exchange of water than the original simulation and the other three strategies. s_{min} is also larger for the Strategy 1 simulation, indicating that there is an increase in residence time along the whole vertical axis through the outlet. Δs is smaller, meaning that the vertical variation of the residence time at the outlet is smaller. This is advantageous since it can be assumed that the largest part of the water being released during the draining process will be part of the 'oldest' water in the reservoir, and that is a sign of good overall exchange of water in the reservoir.

4.2.3 Residence time at outlet

Figure 4.2.3 shows the simulated average residence time at the outlet during the mixing process, for the original and the alternative strategies.

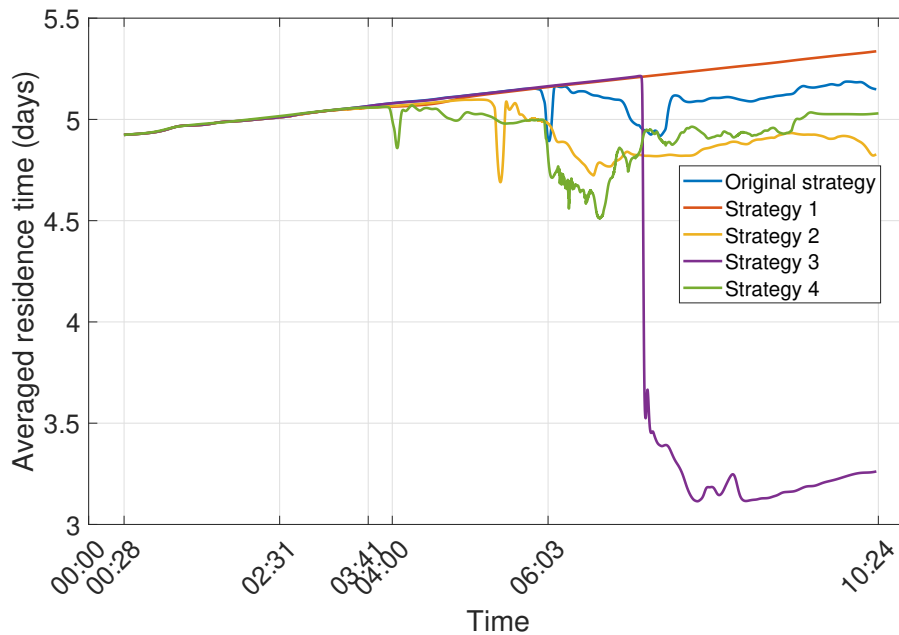


Figure 4.2.3: Simulated average residence time at the outlet as a function of time.

Once again, it can be seen that Strategy 1 is proven to generate the best exchange of water. It has the highest average residence time of water at the outlet towards the end of the mixing process. In other words, the water reaching the outlet during the mixing process of Strategy 1 is 'older' than it is for the other strategy cases.

It is clear that Strategy 3 stands out by having a much lower average residence time at the outlet by the end of the mixing process. It can be reasoned that this is due to

the colder inflow of water which contributes to the creation of stagnant zones due to the 'newer' incoming water not mixing well with the initial reservoir water by directly sinking to the bottom of the reservoir (due to its higher density).

The only way Strategy 2 differed from Strategy 1 was by having a larger water level increase during the filling process. By looking at the yellow curve which shows the average residence time at the outlet for Strategy 2 during the whole mixing process, it can be reasoned that a larger volume of water (higher water level) in the reservoir results in a lower exchange of water. This can be explained by the fact that the water which is located close to the outlet has a lower average residence time compared to Strategy 1 and also compared to the existing maximum residence time in the reservoir.

4.2.4 Temperature at outlet

Figure 4.2.4 shows the simulated average temperature at the outlet, for the original and the alternative strategies.

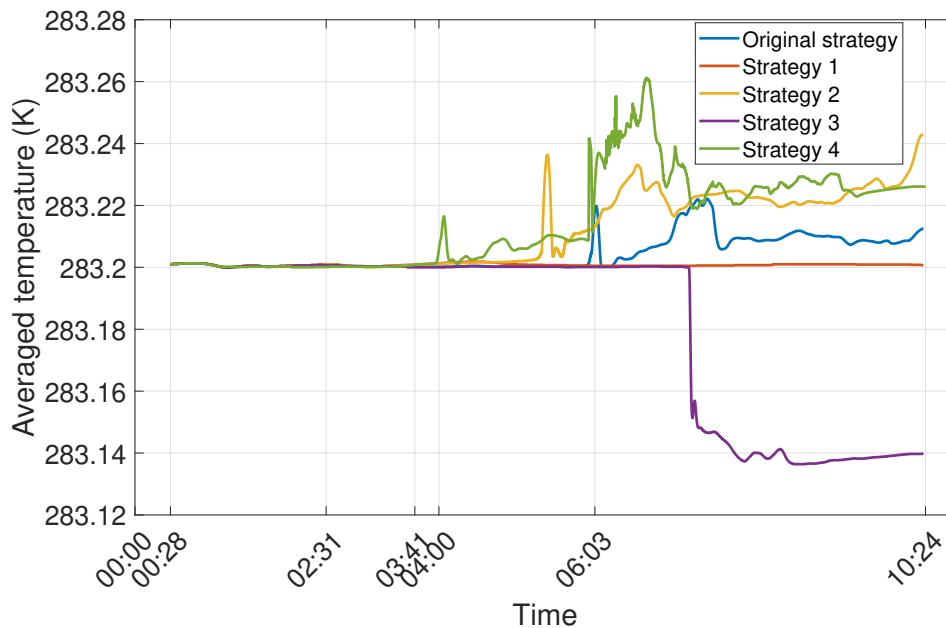


Figure 4.2.4: Simulated average temperature at the outlet as a function of time.

It can be seen that the average temperature variation for Strategy 1 is constant at the outlet, compared to the other cases. A constant average temperature at the outlet during a filling process indicates that there either is not much variation of temperature in the reservoir water or that the water located at the bottom of the reservoir adjacent

to the outlet is stagnant during the entire mixing process. In this case, the latter explanation is more reasonable since it can justify the linearly increasing average residence time at the outlet seen in Figure 4.2.3.

Chapter 5

Discussion

In this chapter, the results will be discussed in relation to literature and flow cytometer data measured by Norrvatten. The choice of method and the strengths and weaknesses of the project work will also be discussed.

5.1 Limitations

The CFD modeling of the reservoir is based on several assumptions. One of them is that the walls are fully isolated meaning that the water temperature in the reservoir is not affected by the outside temperature. In a real-world scenario, one has to consider the heat transfer from outside, especially those days the sun is shining.

One may also question how accurate the simulation of the water temperature in the reservoir is. Since the initial temperature conditions used in the CFD simulation are guessed based on uncertain temperature measurements, it is hard to verify their accuracy. The temperature measurements that are used are taken in the outer pipe of the reservoir (where both inflowing and outflowing water passes) and are therefore only measuring the temperature in either ingoing or outgoing water. Consequently, there is no way to accurately estimate the initial temperature of the water inside the reservoir. This temperature (which is set as an initial temperature condition in the CFD model) is guessed based on the average temperature of incoming water from an earlier filling process. The CFD study would have been more accurate if temperature measurements could have been taken from different heights in the reservoir. Therefore, a solution could be to have several temperature gauges placed

at different heights in the reservoir.

Another disadvantage of not being able to take temperature measurements from different heights in the reservoir is that it makes it impossible to determine the vertical temperature distribution from experimental data. Consequently, the simulated vertical temperature distribution (thermal stratification) can not be validated against experimental data.

Furthermore, the analysis of the water quality in relation to the operational strategies used is restricted due to the BactoSense flow cytometer being located in the external sampling tap (outer pipe). The flow cytometer only measures the microbial activity of water either leaving or entering the reservoir depending on whether the inlet or outlet is open. Therefore, it is hard to get an overall view of the microbial growth in the entire reservoir.

Lastly, the most restrictive limitation of this project was the time constraint. The CFD simulations were simulated in real-time (transient) and required small time steps to accurately resolve the flow conditions ($Co \leq 1$). Since long periods of time were simulated (up to seven days), the simulations were time-consuming. Even though measures were taken to speed up the simulations, they were still very slow.

5.2 Failure analysis

During the CFD study, several simulations crashed due to reasons that were not related to the setup of the solver or the boundary conditions. It was concluded that the majority of the simulation failures were due to an unstable mesh. The mesh became unstable due to the poor quality of some mesh cells. When running the *checkMesh* utility in OpenFOAM it was discovered that some of the cells were consisting of an unreasonably large number of faces (up to 69 faces per cell). The problem with an unstable mesh mostly occurred when the dynamic mesh was shrinking (due to a decrease in water level) and the water level was decreasing below two meters. It seemed as if the quality of the mesh cells were particularly degraded when they were shrinking, which is understandable.

There were no resources available for solving the unstable mesh problem, hence, it was dealt with by going around the problem to avoid a simulation crash. Since simulation failures of this type are quite unpredictable, random adjustments were

made to the mesh to avoid the problem. The problem with the unstable mesh could have been solved if access to the CAD geometry (surface data file in STL format) of the reservoir model (created by DHI) was available. Then another mesh utility called the *snappyHexMesh* could have been used on the original mesh created with the *blockMesh* utility (which is used as a background mesh of hexahedral cells). With *snappyHexMesh* the mesh could have been better adapted to the geometry. By iteratively refining the original mesh and morphing the resulting split-hex mesh to the surface the mesh utility would provide a mesh of higher quality [45].

5.3 Comparison with flow cytometer data

One of the goals of this study was to investigate the correlation between microbial growth and the exchange of water in the reservoir. Flow cytometer data for the period 22-28 September 2021 is presented in Appendix C.2. This data can be compared with the simulation results of the filling processes from the 24th to 28th September, to investigate if a correlation between the microbial activity and the residence time of water exists.

The peaks that appear in the TCC and HNAC curve during the 27th and 28th September can be explained as 'older' water reaching the outlet. The peaks appear both during the filling process and at the start of the draining process. By studying the residence time distributions at the outlet after the filling processes of 27th and 28th September in Figure 4.1.2 in section 4.1.2, it can be seen that the bottom layer of the reservoir contains older water (higher residence time). This may be because the inflowing water of both of the filling processes has a higher temperature than the initial water temperature in the reservoir. Older water which is colder will then sink to the bottom of the reservoir. The peaks in the TCC/HNAC curve can be related to the cultivation results of the water sample that was taken at 10:45 the 28th September which showed an increase in the growth of slow-growing bacteria (see Table 3.3.2 in section 3.3.1). This indicates that an increase in microbial growth may be due to an inadequate exchange of water a few days before the water sample was taken. In this case, the exchange of water was lower on the days with colder inflow (24th and 25th September) which is 3-4 days before the water sample is taken.

However, it is difficult to identify problematic operational strategies by just studying the flow cytometer data since the increase in total cell count may be due to

uncontrollable circumstances in combination with reservoir operation. To gain a deeper understanding of how the operational strategies (inflow rate, water level, and time intervals for the mixing processes) can be improved to decrease the microbial growth in the reservoir, one needs to run more simulations of cases with inflow conditions that are more distinguishable from each other. For example, running a case that has a high water level variation (7-9 meters) and then comparing it to a similar case that has a low water variation (3-5 meters). In addition, it is difficult to find a correlation between cultivation results (which are specified in colony-forming units (CFUs) per ml) and flow cytometer data which measures the total amount of cells per ml. A colony-forming unit (CFU) is a unit used to estimate the number of viable bacteria cells in a sample. It is based on the assumption that each colony arises from a single viable cell. However, it is not possible to accurately determine the exact number of bacteria cells in a CFU, because a colony can arise from a single cell that undergoes many rounds of cell division, producing a large number of cells. Additionally, some bacteria may not grow well in the conditions used for culturing, leading to an underestimation of the number of viable cells. Therefore, a CFU count provides an estimate of the number of viable cells in a sample, but it is not a precise measurement of the exact number of cells [9].

5.4 Exchange of water

The movement and exchange of water in the reservoir depend on several factors, other than the inflow conditions they depend on the outside temperature conditions. The exchange of water is strongly dependent on the temperature variation of the water. If a stable thermal stratification is created, this will most certainly decrease the exchange of water, thus having a negative effect.

In the cases regarding the 27th and 28th September, the residence time distributions shown in Figure 4.1.2 show that the water which has reached the outlet is the same water that entered the reservoir during the first 2-3 days of the simulation week. At the end of the filling process of 27th September the water from the 1st and 2nd filling has reached the outlet, and at the end of the filling process of 28th September the water from the 2nd and 3rd filling has reached the outlet. Results of the simulations show that it takes up to five days for incoming water to reach the outlet after entering the reservoir.

Furthermore, the simulations showed that it is advantageous to keep a low water level in the reservoir during the mixing processes. However, this would not be the case for a cylindrical reservoir without internal walls and sections. Instead, it would probably be beneficial to have a higher water level. This can be justified by the properties of a turbulent jet stream which enables the volume flow rate to increase with distance from the inlet pipe if the stream is allowed to develop freely in the reservoir [21]. Since the results of the simulations showed that it is better to keep a low water level, it can be argued that the inner walls of the reservoir act as obstacles that slow down the jet stream and prevents the volume flow from increasing in the reservoir.

Chapter 6

Conclusions

This study aimed to identify the optimal operational strategies of a reservoir run by Norrvatten by investigating how the flow conditions could affect the exchange of water. Moreover, the aim was to investigate the relationship between the residence time of the water and the microbial growth in the reservoir. How the operational strategies relate to the residence time and microbial growth is also of interest, as well as the effect of the thermal stratification.

Based on the simulations of the previous operational strategies, it can be concluded that the temperature distribution is strongly related to the vertical distribution of residence time. Mixing processes with colder inflows are problematic since they produce a thermal stratification with 'newer' water at the bottom of the reservoir close to the outlet. These cases have been proven to have a lower exchange of water. However, a long mixing process can help even out the temperature variations in the vertical direction, and thereby contribute to a smaller residence time variation. This may slightly increase the mixing of the water. The difference between the temperature of incoming water and the initial temperature in the reservoir ($T_0 - T_a$) has a significant impact on the creation of thermal stratification. Other than the quite obvious conclusions, it is difficult to draw any conclusions about what specific strategies are favorable when comparing the simulations to each other.

Based on the simulations of the alternative operational strategies, it can be concluded that 'Strategy 1' is the most advantageous strategy for reservoir operation considering all aspects evaluated. Compared to the other simulations, the mixing process simulation with Strategy 1 was shown to have the smallest temperature variation at

the outlet by the end of the mixing process, which indicates a less prominent thermal stratification. It was also proven to have the highest average residence time of water at the outlet by the end of the mixing process. During the mixing process, the average residence time at the outlet increases linearly while the average outlet temperature remains constant. This behavior nearly resembles the flow conditions of a plug flow, and it can therefore be assumed that the operational strategies of Strategy 1 are the most suitable for this specific reservoir due to its snail-shaped geometry.

The favorable operational strategies that were identified according to Strategy 1 are:

- Higher volumetric inflow rate (compared to original strategy)
- Lower water level (compared to Strategy 2)
- Temperature of incoming water is higher than the initial temperature in the reservoir (compared to Strategy 3)

Based on the assumption that the exchange of water in the reservoir can be assessed using the average residence time variation at the outlet (\bar{s}_{outlet}) during an entire mixing process, the difference in exchange of water of the different strategies can be evaluated. Compared to the original strategy, the Strategy 1 approach is proven to increase the average residence time (at the outlet at the last time step of the mixing process) by 3.6 %. According to the previously stated assumption, this corresponds to a 3.6 % increase in the exchange of water. Compared to the Strategy 2 approach, the Strategy 1 approach generates a 10.6 % higher exchange of water, and compared to the Strategy 3 approach, the Strategy 1 approach generates a 63.6 % higher exchange of water. It is proven that the worst-case scenario when operating the reservoir is to have a colder inflow of water, as the Strategy 3 approach is based on. A decrease of 0.726 K in the temperature of incoming water (0.22 K colder than the initial temperature in the reservoir) results in a 36.7 % decrease in the exchange of water.

6.1 Future work

Based on these conclusions and the limitations of the project, recommended future work can be to:

- Measure the temperature at different heights in the reservoir to get some experimental data on the temperature distribution. CFD simulations could then

be validated against this data. If the simulations can predict the temperature distribution reasonably correctly, they are accurate.

- Improve the quality of the mesh, for example by using *snappyHexMesh*, to avoid simulation failures and to make the simulations more accurate. Additionally, this would enable the simulation of a complete draining of the reservoir.

Bibliography

- [1] B. Berghult, O. Bergstedt, and e. al., "Dricksvattenteknik 4 Efterbehandling och distribution," *U9*, p. 182, Mar. 2011.
- [2] "Dricksvattenproduktion." [Online]. Available: <https://www.norrvatten.se/dricksvatten/dricksvattenproduktion/>
- [3] "Rationell mikrobiell analys för biostabilt dricksvatten." [Online]. Available: <https://www.svensktvatten.se/forskning/sa-jobbar-vi-med-forskning-svu/pagaende-svu-projekt2/rationell-mikrobiell-analys-for-biostabilt-dricksvatten/>
- [4] M. Danielsson, L. Holmer, D. Hellström, C. Schleich, A. Keucken, J. Barup, L. Meyer-Lind, S. Chan, T. Rosenqvist, C. J. Paul, and P. Rådström, "Mikrobiologisk analys för biostabilt dricksvatten," © Svenskt Vatten AB, Tech. Rep. 2022-9, 2022.
- [5] "Svenskt Vatten," Jul. 2022. [Online]. Available: <https://www.svensktvatten.se/om-oss/in-english/>
- [6] "Livsmedelsverkets föreskrifter (SLVFS 2001:30) om dricksvatten;." [Online]. Available: https://www.livsmedelsverket.se/globalassets/om-oss/lagstiftning/nummerordning---upphord-lagstiftning/2001/slvfs-2001-30-hela_foreskriften.pdf
- [7] C. Schleich, S. Chan, K. Pullerits, M. Habagil, J. Lindgren, C. J. Paul, A. Keucken, and P. Rådström, "Biofilmens funktion och korrelation med dricksvattnets kvalitet," *Svenskt Vatten*, no. 2020-2, p. 54. [Online]. Available: <https://vattenbokhandeln.svensktvatten.se/produkt/biofilmens-funktion-och-korrelation-med-dricksvattnets-kvalitet/>
- [8] H. Annadotter, B. Berghult, O. Bergstedt, A. Blom, H. Danielsson, A. Elfström Broo, M. Engdahl, U. Eriksson, U. Eriksson, C. Frycklund, G. Heinicke, T. Helmerson, T. Holm,

- R. Kling, S.-E. Kristenson, C.-J. Legeth, K. M Persson, H. Rydberg, E. Sandgren, and K. Törneke, "Dricksvattenteknik 1 Vatten i natur och samhälle," *U6*, p. 105, Dec. 2010.
- [9] E. Goldman and L. H. Green, *Practical Handbook of Microbiology*. CRC Press, Aug. 2008.
- [10] J. T. Staley and A. Konopka, "Measurement of in Situ Activities of Nonphotosynthetic Microorganisms in Aquatic and Terrestrial Habitats," *Annual Review of Microbiology*, vol. 39, no. 1, pp. 321–346, 1985. [Online]. Available: <https://doi.org/10.1146/annurev.mi.39.100185.001541>
- [11] F. Hammes, M. Berney, Y. Wang, M. Vital, O. Köster, and T. Egli, "Flow-cytometric total bacterial cell counts as a descriptive microbiological parameter for drinking water treatment processes," *Water Research*, vol. 42, no. 1, pp. 269–277, Jan. 2008. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0043135407004691>
- [12] E. Gatza, F. Hammes, and E. Prest, "Assessing Water Quality with the BD Accuri™ C6 Flow Cytometer," p. 12, 2013.
- [13] S. A. Bates, "Nucleic Acids." [Online]. Available: <https://www.genome.gov/genetics-glossary/Nucleic-Acids>
- [14] "Flödescytometri på Norrvatten," Apr. 2022. [Online]. Available: <https://www.norrvatten.se/dricksvatten/forskningsprojekt/flodescytometri/>
- [15] "BactoSense | Online Monitoring." [Online]. Available: <https://www.bnovate.com/bactosense>
- [16] T. Manickum, *Total Colony Counts (TCC) By Flow Cytometry (FCM) Should Replace the Heterotrophic Plate Count (HPC) Test for Bacteriological Enumeration of Water -Some Recent Developments in Flow Cytometry: A Review*, Jul. 2020.
- [17] P. Lebaron, P. Servais, H. Agogué, C. Courties, and F. Joux, "Does the High Nucleic Acid Content of Individual Bacterial Cells Allow Us To Discriminate between Active Cells and Inactive Cells in Aquatic Systems?" *Applied and Environmental Microbiology*, vol. 67, no. 4, pp. 1775–1782, Apr. 2001, tex.pmcid: PMC92796. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC92796/>
- [18] J. M. Gasol and P. A. d. Giorgio, "Using flow cytometry for counting

- natural planktonic bacteria and understanding the structure of planktonic bacterial communities,” *Scientia Marina*, vol. 64, no. 2, pp. 197–224, Jun. 2000, tex.copyright: Copyright (c) 2000 Consejo Superior de Investigaciones Científicas (CSIC). [Online]. Available: <https://scientiamarina.revistas.csic.es/index.php/scientiamarina/article/view/755>
- [19] L. Bernard, C. Courties, P. Servais, M. Troussellier, M. Petit, and P. Lebaron, “Relationships among Bacterial Cell Size, Productivity, and Genetic Diversity in Aquatic Environments using Cell Sorting and Flow Cytometry,” *Microbial Ecology*, vol. 40, no. 2, pp. 148–158, Aug. 2000. [Online]. Available: <http://link.springer.com/10.1007/s002480000046>
- [20] S. B. Pope, *Turbulent flows*. Cambridge, United Kingdom: University of Cambridge, 2000.
- [21] O. Nordblom and O. Ljunggren, “Blandning och omsättning i dricksvattenreservoarer,” Svenskt Vatten AB, Tech. Rep. 2007-15, 2007.
- [22] “Plug flow,” Mar. 2019, tex.copyright: Creative Commons Attribution-ShareAlike License. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Plug_flow&oldid=888808696
- [23] O. Nordblom and L. Bergdahl, “Initiation of Stagnation in Drinking Water Storage Tanks,” *Journal of Hydraulic Engineering*, vol. 130, no. 1, pp. 49–57, Jan. 2004. [Online]. Available: <https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9429%282004%29130%3A1%2849%29>
- [24] R. Eymard, Herbin, and T. Gallouët, “Finite volume method,” *Scholarpedia*, vol. 5, no. 6, p. 9835, Jun. 2010. [Online]. Available: http://www.scholarpedia.org/article/Finite_volume_method
- [25] L. Davidson, “Fluid mechanics, turbulent flow and turbulence modeling,” p. 585.
- [26] S. Dey, P. Paul, S. Z. Ali, and E. Padhi, “Reynolds stress anisotropy in flow over two-dimensional rigid dunes,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 476, no. 2242, p. 20200638, Oct. 2020. [Online]. Available: <https://royalsocietypublishing.org/doi/10.1098/rspa.2020.0638>
- [27] Fluid Mechanics 101, “[CFD] The k - epsilon Turbulence Model,” Jun. 2019. [Online]. Available: <https://www.youtube.com/watch?v=fOB91zQ7HJU>

- [28] W. P. Jones and B. E. Launder, “The prediction of laminarization with a two-equation model of turbulence,” *International Journal of Heat and Mass Transfer*, vol. 15, no. 2, pp. 301–314, 1972. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0017931072900762>
- [29] J. H. Ferziger, M. Perić, and R. L. Street, *Computational Methods for Fluid Dynamics*, 4th ed. Cham: Springer International Publishing, tex.copyright: Springer Nature Switzerland AG 2020.
- [30] Fluid Mechanics 101, “[CFD] The Boussinesq Approximation for Bouyancy Driven (Natural Convection) Flow,” Oct. 2018. [Online]. Available: <https://www.youtube.com/watch?v=onKiVbKSoXw>
- [31] —, “[CFD] The Energy Equation for Solids and Fluids in CFD,” Nov. 2018. [Online]. Available: https://www.youtube.com/watch?v=z8dZHze_EP0
- [32] C. Greenshields and H. Weller, “Notes on Computational Fluid Dynamics: General Principles,” Apr. 2022. [Online]. Available: <https://doc.cfd.direct/notes/cfd-general-principles/heat-transfer-in-turbulent-flow/>
- [33] “OpenFOAM – CFD-Wiki, the free CFD reference,” Feb. 2019. [Online]. Available: <https://www.cfd-online.com/Wiki/OpenFOAM>
- [34] Fluid Mechanics 101, “[CFD] The PISO Algorithm,” Nov. 2019. [Online]. Available: <https://www.youtube.com/watch?v=ahdW5TKacok>
- [35] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method*, 2nd ed. Harlow, England ; New York: Pearson Education Ltd, 2007.
- [36] T. Holzmann, *Mathematics, numerics, derivations and OpenFOAM®*, Nov. 2019.
- [37] “OpenFOAM: User Guide: CourantNo.” [Online]. Available: <https://www.openfoam.com/documentation/guides/latest/doc/guide-fos-field-CourantNo.html>
- [38] “OpenFOAM: API Guide: buoyantBoussinesqPimpleFoam.C File Reference.” [Online]. Available: https://www.openfoam.com/documentation/guides/latest/api/buoyantBoussinesqPimpleFoam_8C.html#details
- [39] “Law of the wall,” May 2022, tex.copyright: Creative Commons Attribution-ShareAlike License. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Law_of_the_wall&oldid=1087128201

- [40] Fluid Mechanics 101, “[CFD] What are Wall Functions and How do they work?” Sep. 2018. [Online]. Available: <https://www.youtube.com/watch?v=fJDYtEGMgzs>
- [41] —, “[CFD] What is the difference between y^+ and y^* ?” Apr. 2019. [Online]. Available: <https://www.youtube.com/watch?v=nSdVaF3JnIo>
- [42] C. L. V. Jayatilleke, “THE INFLUENCE OF PRANDTL NUMBER AND SURFACE ROUGHNESS ON THE RESISTANCE OF THE LAMINAR SUB-LAYER TO MOMENTUM AND HEAT TRANSFER.” phd, Imperial College of Science and Technology, 1969.
- [43] “Turbulence free-stream boundary conditions – CFD-Wiki, the free CFD reference.” [Online]. Available: https://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions
- [44] C. Greenshields, “OpenFOAM v6 User Guide,” Jul. 2018. [Online]. Available: <https://doc.cfd.direct/openfoam/user-guide-v6/fvsolution/>
- [45] “Mesh generation with the snappyHexMesh utility.” [Online]. Available: <https://www.openfoam.com/documentation/user-guide/4-mesh-generation-and-conversion/4-mesh-generation-with-the-snappyhexmesh-utility>

Appendix - Contents

A Theory	100
A.1 Boundary layer treatment	100
A.2 The reservoir	101
B OpenFOAM code	102
B.1 The system directory	102
B.1.1 Mesh generation	102
B.1.2 Inlet and outlet	106
B.1.3 Vent implementation	108
B.1.4 Trace element implementation	109
B.2 The 0 directory	109
B.2.1 Velocity	109
B.2.2 Temperature	111
B.2.3 Pressure	113
B.2.4 Trace element	113
B.2.5 Point motion of water level	114
B.2.6 Turbulent viscosity	116
B.2.7 Turbulent kinetic energy	117
B.2.8 Dissipation of turbulent kinetic energy	118
B.2.9 Turbulent thermal diffusivity	119
C Post-processing	120
C.1 Temperature distributions at outlet	120
C.1.1 Alternative simulations	124
C.2 Flow cytometer data	126

Appendix A

Theory

A.1 Boundary layer treatment

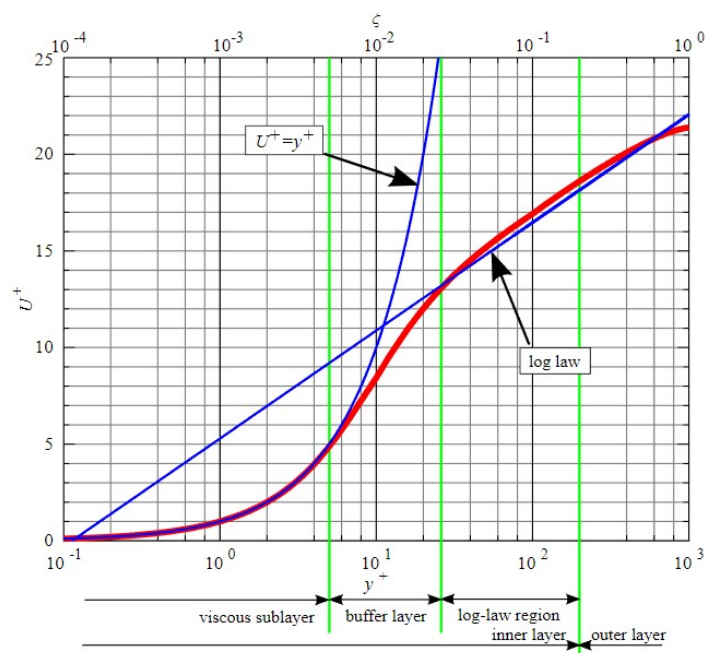


Figure A.1.1: Law of the wall. [39]

A.2 The reservoir

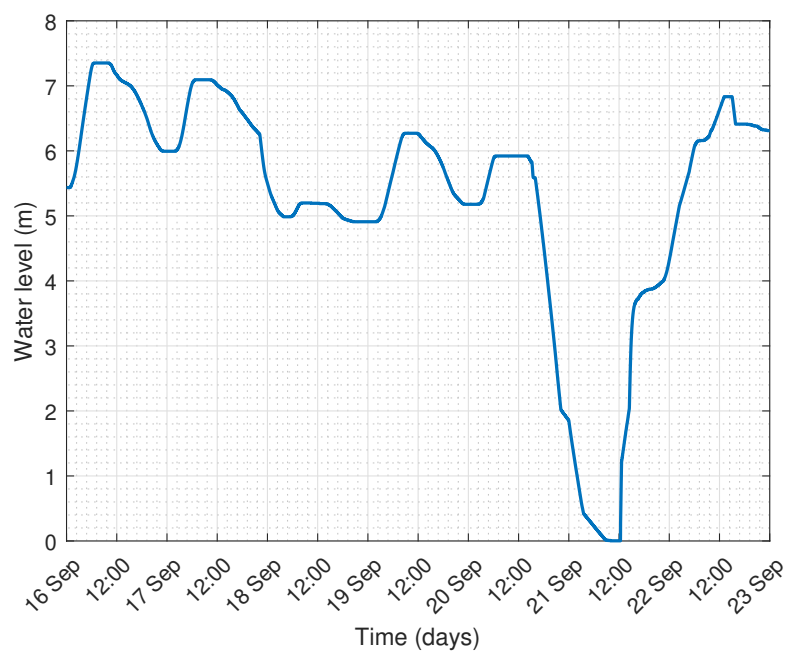


Figure A.2.1: The water level as function of time in the reservoir during the period 16th to 23rd September 2021.

Appendix B

OpenFOAM code

B.1 The system directory

B.1.1 Mesh generation

```
convertToMeters 1;

// radius
// r1=3250 mm
// r2=10800 mm
// r3=11000 mm
// r4=18175 mm

a 5.3982; // initial water depth

i1 129; //70 no of cells in flow direction, inner circle, box 0
i2 92; //50 no of cells in flow direction, inner circle, box 1
i3 183; //100 no of cells in flow direction, outer circle, box 4
i4 146; //80 no of cells in flow direction, outer circle, box 5
j1 45; //25 no of cells in cross flow direction, inner circle, box 0,1,2
j2 37; //20 no of cells in cross flow direction, outer circle, box 3,4,5,6
k 56; //30 no of cells in vertical direction
m 65; //35 no of cells along straight wall, inner and outer circle, box 2,6
```

```
vertices
(
  (-2.815 -1.625 0) //0
  (3.25 0 0)
  (10.8 0 0) //2
  (-10.129 3.746 0)
  //
  (-2.815 -1.625 $a) //4
  (3.25 0 $a)
  (10.8 0 $a) //6
  (-10.129 3.746 $a)
  //
  (-2.001 -2.561 0) //8
  (-6.649 -8.511 0)
  (-2.001 -2.561 $a) //10
  (-6.649 -8.511 $a)
  //
  (-10.369 3.672 0) //12
  (-15.074 -1.75 0)
  (-10.369 3.672 $a) //14
  (-15.074 -1.75 $a)
  //
  (-16.336 7.967 0) //16
  (-18.051 -2.116 0)
  (-16.336 7.967 $a) //18
  (-18.051 -2.116 $a)
  //
  (11 0 0) // 20
  (18.175 0 0)
  (11 0 $a) // 22
  (18.175 0 $a)
  //
  (-6.772 -8.668 0) // 24
  (-11.19 -14.322 0)
  (-6.772 -8.668 $a) // 26
  (-11.19 -14.322 $a)
  //
  (-15.049 -1.949 0) // 28
  (-18.027 -2.314 0)
  (-15.049 -1.949 $a) // 30
  (-18.027 -2.314 $a)
);
```

```
blocks
(
    hex (0 1 2 3 4 5 6 7) ($i1 $j1 $k) simpleGrading (1 1 1)
    hex (1 8 9 2 5 10 11 6) ($i2 $j1 $k) simpleGrading (1 1 1)
    hex (9 8 12 13 11 10 14 15) ($j1 $m $k) simpleGrading (1 1 1)
    hex (13 12 16 17 15 14 18 19) ($j1 $j2 $k) simpleGrading (1 1 1)
    hex (12 20 21 16 14 22 23 18) ($i3 $j2 $k) simpleGrading (1 1 1)
    hex (20 24 25 21 22 26 27 23) ($i4 $j2 $k) simpleGrading (1 1 1)
    hex (25 24 28 29 27 26 30 31) ($j2 $m $k) simpleGrading (1 1 1)
    //
);

edges
(
    arc 0 1 (0 3.250 0) //
    arc 4 5 (0 3.250 $a) //
    //
    arc 2 3 (0 10.800 0) //
    arc 6 7 (0 10.800 $a) //
    //
    arc 1 8 (0 -3.250 0) //
    arc 5 10 (0 -3.250 $a) //
    //
    arc 9 2 (0 -10.800 0) //
    arc 11 6 (0 -10.800 $a) //
    //
    arc 17 16 (-18.175 0 0) //
    arc 19 18 (-18.175 0 $a) //
    //
    arc 12 20 (0 11 0) //
    arc 14 22 (0 11 $a) //
    //
    arc 16 21 (0 18.175 0) //
    arc 18 23 (0 18.175 $a) //
    //
    arc 20 24 (0 -11 0) //
    arc 22 26 (0 -11 $a) //
    //
    arc 21 25 (0 -18.175 0) //
    arc 23 27 (0 -18.175 $a) //
    //
    arc 25 29 (-15.74 -9.0875 0) //
    arc 27 31 (-15.74 -9.0875 $a) //
```

```
boundary
(
    walls_vertical
    {
        type wall;
        faces
        (
            (0 1 5 4)
            (2 3 7 6)
            (3 0 4 7)
            (1 8 10 5)
            (9 2 6 11)
            (8 12 14 10)
            (13 9 11 15)
            (17 13 15 19)
            (16 17 19 18)
            (12 20 22 14)
            (21 16 18 23)
            (20 24 26 22)
            (25 21 23 27)
            (24 28 30 26)
            (29 25 27 31)
            (28 29 31 30)
        );
    }
    walls_floor
    {
        type wall;
        faces
        (
            (0 3 2 1)
            (1 2 9 8)
            (8 9 13 12)
            (13 17 16 12)
            (12 16 21 20)
            (20 21 25 24)
            (24 25 29 28)
        );
    }
}

waterlevel
{
    type wall;
    faces
    (
        (7 4 5 6)
        (6 5 10 11)
        (11 10 14 15)
        (19 15 14 18)
        (18 14 22 23)
        (23 22 26 27)
        (27 26 30 31)
    );
}

defaultFaces
{
    type wall;
    faces
    (
    );
}

);

mergePatchPairs
(
);
```

Figure B.1.1: Source code from the *blockMeshDict* file.

B.1.2 Inlet and outlet

```
actions
(
    // *****
    // INLET
    // *****
    {
        name c0;
        type cellSet;
        action new;
        source cylinderToCell;
        sourceInfo
        {
            p1      (-15.77 -3.35 -1);
            p2      (-15.77 -3.35 1);
            radius   0.3;
        }
    }

    {
        name inletFaces;
        type faceSet;
        action new;
        source cellToFace;
        sourceInfo
        {
            set c0;
            option all;
        }
    }

    {
        name inletFaces;
        type faceSet;
        action subset;
        source patchToFace;
        sourceInfo
        {
            name "walls_floor";
        }
    }

    {
        name c0;
        type cellSet;
        action remove;
    }
)
```



```
// *****
// OUTLET
// *****
{
    name c0;
    type cellSet;
    action new;
    source cylinderToCell;
    sourceInfo
    {
        p1      (-5.29 1.54 -1);
        p2      (-5.29 1.54 1);
        radius   0.3;
    }
}

{
    name outletFaces;
    type faceSet;
    action new;
    source cellToFace;
    sourceInfo
    {
        set c0;
        option all;
    }
}

{
    name outletFaces;
    type faceSet;
    action subset;
    source patchToFace;
    sourceInfo
    {
        name "walls_floor";
    }
}

{
    name c0;
    type cellSet;
    action remove;
}
_ }
```

Figure B.1.2: Source code for the inlet and outlet implementation from the *topoSetDict* file.

B.1.3 Vent implementation

```
// *****  
// VENT  
// *****  
{  
    name c0;  
    type cellSet;  
    action new;  
    source cylinderToCell;  
    sourceInfo  
    {  
        p1      (0 13 -1);  
        p2      (0 13 1);  
        radius   0.2;  
    }  
}  
  
{  
    name ventFaces;  
    type faceSet;  
    action new;  
    source cellToFace;  
    sourceInfo  
    {  
        set c0;  
        option all;  
    }  
}  
  
{  
    name ventFaces;  
    type faceSet;  
    action subset;  
    source patchToFace;  
    sourceInfo  
    {  
        name "walls_floor";  
    }  
}  
  
{  
    name c0;  
    type cellSet;  
    action remove;  
}
```

Figure B.1.3: Source code from the *topoSetDict* file.

B.1.4 Trace element implementation

```
S
{
    type            scalarTransport;
    libs ("libsolverFunctionObjects.so")
    writeControl     outputTime;
    writeInterval    1;
    DT               14e-6;
    userDT           false;
    resetOnStartup   false;
    autoSchemes      true;
    //schemesField    k;
    fvOptions
    {
        IncrementTime
        {
            //type semiImplicitSource;
            type scalarSemiImplicitSource
            active      true;
            selectionMode all;
            volumeMode  specific;
            injectionRateSuSp
            {
                s (1 0);
            }
        }
    }
};
}
```

Figure B.1.4: Source code for the trace element implementation from the *controlDict* file.

B.2 The 0 directory

B.2.1 Velocity

```
dimensions      [0 1 -1 0 0 0 0];
internalField    uniform (0 0 0);

boundaryField
{
    "(walls.*)"
    {
        type            fixedValue;
        value            uniform (0 0 0);
    }

    waterlevel
    {
        type            myMovingWallSlip;
        value            $internalField;
    }
}
```

```
inlet
{
    type flowRateInletVelocity;
    volumetricFlowRate
    table
    (
        (0 0.0)
        (2280 0)
        (2340 0.0561)
        (22259 0.0561)
        (22260 0)
        (89820 0)
        (89880 0.02)
        (143039 0.02)
        (143040 0)
        (186540 0)
        (186600 0.0631)
        (203699 0.0631)
        (203700 0)
        (282480 0)
        (282540 0.0622)
        (291179 0.0622)
        (291180 0)
        (335790 0)
        (335820 0.1015)
        (404219 0.1015)
        (404220 0)
        (440940 0)
        (441000 0.0546)
        (453539 0.0546)
        (453540 0)
        (523680 0)
        (523740 0.0325)
        (572519 0.0325)
        (572520 0)
    );
}

outlet
{
    type flowRateInletVelocity;
    volumetricFlowRate
    table
    (
        (0 0.0)
        (30980 0)
        (31080 -0.0064)
        (79499 -0.0064)
        (79500 0)
        (149851 0)
        (150000 -0.0170)
        (181379 -0.0170)
        (181380 0)
        (291440 0)
        (291540 -0.1515)
        (324000 -0.1515)
        (324001 0)
        (411140 0)
        (411240 -0.0338)
        (430499 -0.0338)
        (430500 0)
        (462320 0)
        (462420 -0.0413)
        (505499 -0.0413)
        (505500 0)
        (581900 0)
        (582000 -0.0253)
        (604800 -0.0253)
    );
}
vent
{
    type zeroGradient;
}
}
```

Figure B.2.1: Source code for the velocity boundary condition from the *U* file.

B.2.2 Temperature

```
dimensions      [0 0 0 1 0 0 0];

internalField    uniform 284.45;

boundaryField
{
    "(walls.*)"
    {
        type      zeroGradient;
    }
    waterlevel
    {
        type      zeroGradient;
    }
    inlet
    {
        type      codedMixed;
        refValue   uniform 284.4455;
        refGradient uniform 0;
        valueFraction uniform 1;
    }

    name      tempMixed;    // name of generated BC

    code
    #{
        // get current time
        scalar t = this->db().time().value();

        if ((t > 0) && (t <= 2280) || (t >= 22260) && (t <= 89820)
            || (t >= 143040) && (t <= 186540) || (t >= 203700) && (t <= 282480)
            || (t >= 291180) && (t <= 335790) || (t >= 404220) && (t <= 440940)
            || (t >= 453540) && (t <= 523680) || (t >= 572520) && (t <= 604800))
        {
            this->valueFraction() = 0;          //zeroGradient
        }
        else if ((t > 2280) && (t < 22260) || (t > 89820) && (t < 143040)
            || (t > 186540) && (t < 203700) || (t > 282480) && (t < 291180)
            || (t > 335790) && (t < 404220) || (t > 440940) && (t < 453540)
            || (t > 523680) && (t < 572520))
        {
            this->valueFraction() = 1;          //fixedValue
        }
    }
```

```
    if ((t > 2280) && (t < 22260))
    {
        this->refValue() = 284.4455;
    }
    else if ((t > 89820) && (t < 143040))
    {
        this->refValue() = 284.4436;
    }
    else if ((t > 186540) && (t < 203700))
    {
        this->refValue() = 284.1428;
    }
    else if ((t > 282480) && (t < 291180))
    {
        this->refValue() = 284.3936;
    }
    else if ((t > 335790) && (t < 404220))
    {
        this->refValue() = 284.1868;
    }
    else if ((t > 440940) && (t < 453540))
    {
        this->refValue() = 284.211;
    }
    else if ((t > 523680) && (t < 572520))
    {
        this->refValue() = 283.9186;
    }
}

outlet
{
    type            zeroGradient;
}
vent
{
    type            zeroGradient;
}
```

Figure B.2.2: Source code for the temperature boundary condition from the T file.

B.2.3 Pressure

```
dimensions      [0 2 -2 0 0 0 0];
internalField    uniform 0;
boundaryField
{
    "(walls.*|inlet|waterlevel|outlet)"
    {
        type      fixedFluxPressure;
        rho        rhok;
        value      uniform 0;
    }

    vent
    {
        type      fixedValue;
        rho        rhok;
        value      uniform 0;
    }
}
```

Figure B.2.3: Source code for the pressure boundary condition from the *p_rgh* file.

B.2.4 Trace element

```
dimensions      [0 0 0 0 0 0 0];
internalField    uniform 0;
boundaryField
{
    inlet
    {
        type      fixedValue;
        value      uniform 0;
    }
    "(walls.*|waterlevel)"
    {
        type      zeroGradient;
    }

    outlet
    {
        type      zeroGradient;
    }
    vent
    {
        type      zeroGradient;
    }
}
```

Figure B.2.4: Source code for the trace element boundary condition from the *s* file.

B.2.5 Point motion of water level

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

internalField    uniform 0;

boundaryField
{
    waterlevel
    {
        type uniformFixedValue;
        uniformValue
        table
        (
            (0 0.0)
            (2280 0)
            (2340 5.6753e-5)
            (22259 5.6753e-5)
            (22260 0)
            (30980 0)
            (31080 -6.4766e-6)
            (79499 -6.4766e-6)
            (79500 0)
            (89820 0)
            (89880 2.0252e-5)
            (143039 2.0252e-5)
            (143040 0)
            (149851 0)
            (150000 -1.7159e-5)
            (181379 -1.7159e-5)
            (181380 0)
            (186540 0)
            (186600 6.3777e-5)
            (203699 6.3777e-5)
            (203700 0)
            (282480 0)
            (282540 6.2868e-5)
            (291179 6.2868e-5)
            (291180 0)
            (291440 0)
            (291540 -1.532e-4)
            (324000 -1.532e-4)
            (324001 0)
            (335790 0)
            (335820 1.0269e-4)
            (404219 1.0269e-4)
        )
    }
}
```



```
(404220 0)
(411140 0)
(411240 -3.4184e-5)
(430499 -3.4184e-5)
(430500 0)
(440940 0)
(441000 5.5204e-5)
(453539 5.5204e-5)
(453540 0)
(462320 0)
(462420 -4.174e-5)
(505499 -4.174e-5)
(505500 0)
(523680 0)
(523740 3.291e-5)
(572519 3.291e-5)
(572520 0)
(581900 0)
(582000 -2.5594e-5)
(604800 -2.5594e-5)
);
}
walls_vertical
{
    type            slip;
}
"(inlet|outlet)"
{
    type            uniformFixedValue;
    uniformValue    constant 0;
}
walls_floor
{
    type            uniformFixedValue;
    uniformValue    constant 0;
}
vent
{
    type            zeroGradient;
}
```

Figure B.2.5: Source code for the point motion boundary condition from the *pointMotionUz* file.

B.2.6 Turbulent viscosity

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

internalField   uniform 0;

boundaryField
{
    "(walls.*)"
    {
        type      nutkWallFunction;
        value      uniform 0;
    }
    waterlevel
    {
        type      calculated;
        value      uniform 0;
        //type      symmetryPlane;
    }
    inlet
    {
        type      calculated;
        value      uniform 0;
    }
    outlet
    {
        type      calculated;
        value      uniform 0;
    }
    vent
    {
        type      zeroGradient;
    }
}
```

Figure B.2.6: Source code for the turbulent viscosity boundary condition from the *nut* file.

B.2.7 Turbulent kinetic energy

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

k 0.003; //k = 1.5*(I*U)**2

internalField   uniform 1e-5;

boundaryField
{
    "(walls.*)"
    {
        type      kqRWallFunction;
        value      uniform $k;
    }
    waterlevel
    {
        type      zeroGradient;
        //type     symmetryPlane;
    }

    inlet
    {
        //type      fixedValue;
        //value      uniform $k;

        type      turbulentIntensityKineticEnergyInlet;
        intensity  0.03;
        value      uniform $k;
    }
    outlet
    {
        type      zeroGradient;
    }
    vent
    {
        type      zeroGradient;
    }
}
```

Figure B.2.7: Source code for the turbulent kinetic energy boundary condition from the *k* file.

B.2.8 Dissipation of turbulent kinetic energy

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

e 0.01; //e = 0.09^0.75*k^1.5/$L

internalField   uniform 1e-5;

boundaryField
{
    "(walls.*)"
    {
        type      epsilonWallFunction;
        value      uniform $e;
    }
    waterlevel
    {
        type      zeroGradient;
    }
    inlet
    {
        type      turbulentMixingLengthDissipationRateInlet;
        mixingLength
        value      0.025; //3-7% of diameter
                  uniform $e;

        //type      fixedValue;
        //value      uniform $e;
    }
    outlet
    {
        type      zeroGradient;
    }
    vent
    {
        type      zeroGradient;
    }
}
```

Figure B.2.8: Source code for the turbulence dissipation rate boundary condition from the *epsilon* file.

B.2.9 Turbulent thermal diffusivity

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

internalField    uniform 0;

boundaryField
{
    "(walls.*)"
    {
        type      alphasJayatillekeWallFunction;
        Prt       0.85;
        value      uniform 0;
    }
    "(inlet|outlet|waterlevel)"
    {
        type      calculated;
        value      uniform 0;
    }
    vent
    {
        type      zeroGradient;
    }
}
```

Figure B.2.9: Source code for the turbulent thermal diffusivity boundary condition from the *alphat* file.

Appendix C

Post-processing

C.1 Temperature distributions at outlet

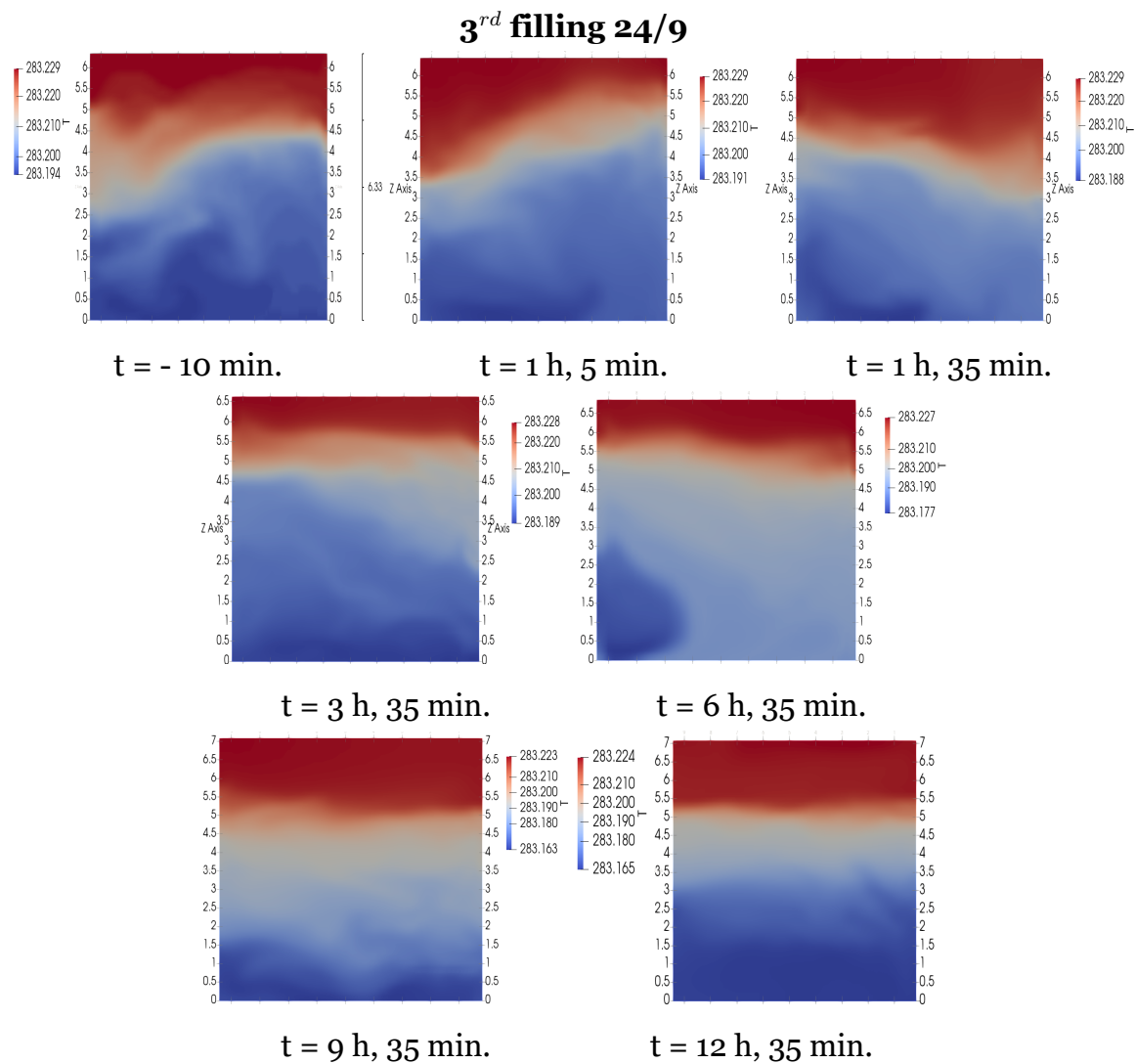


Figure C.1.1: Simulated temperature distribution at the outlet at various times before, during, and after the filling process the 24th September 2021.

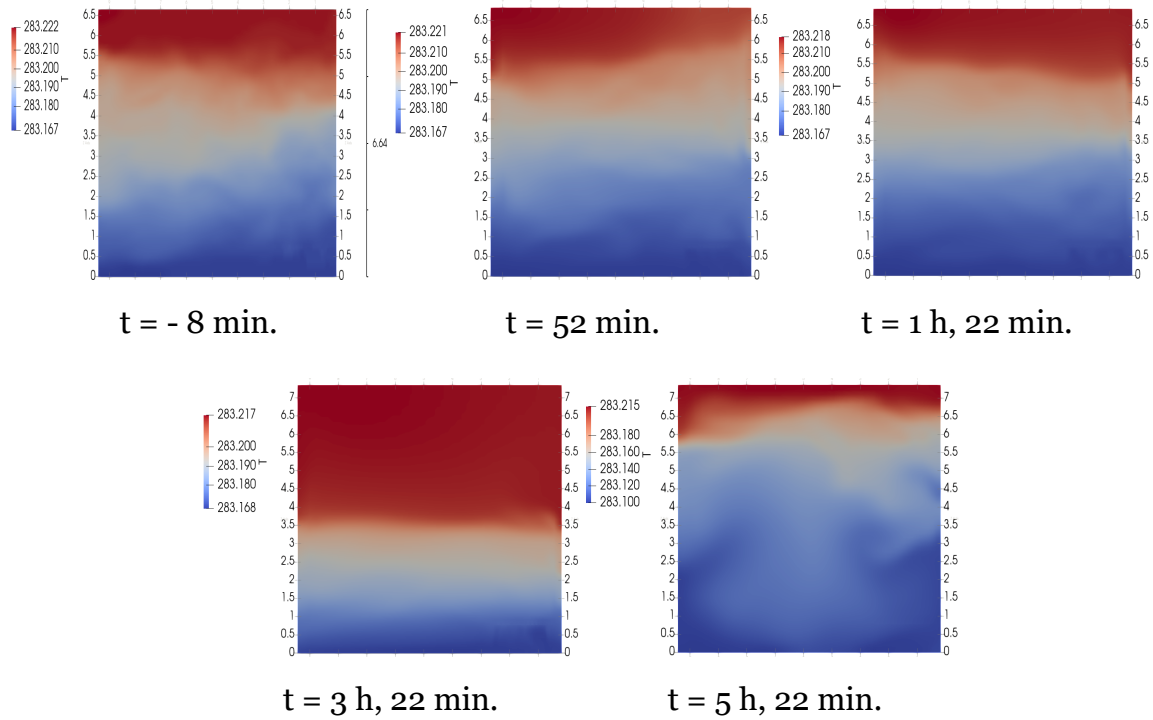
4th filling 25/9

Figure C.1.2: Simulated temperature distribution at the outlet at various times before, during and after the filling process the 25th September 2021.

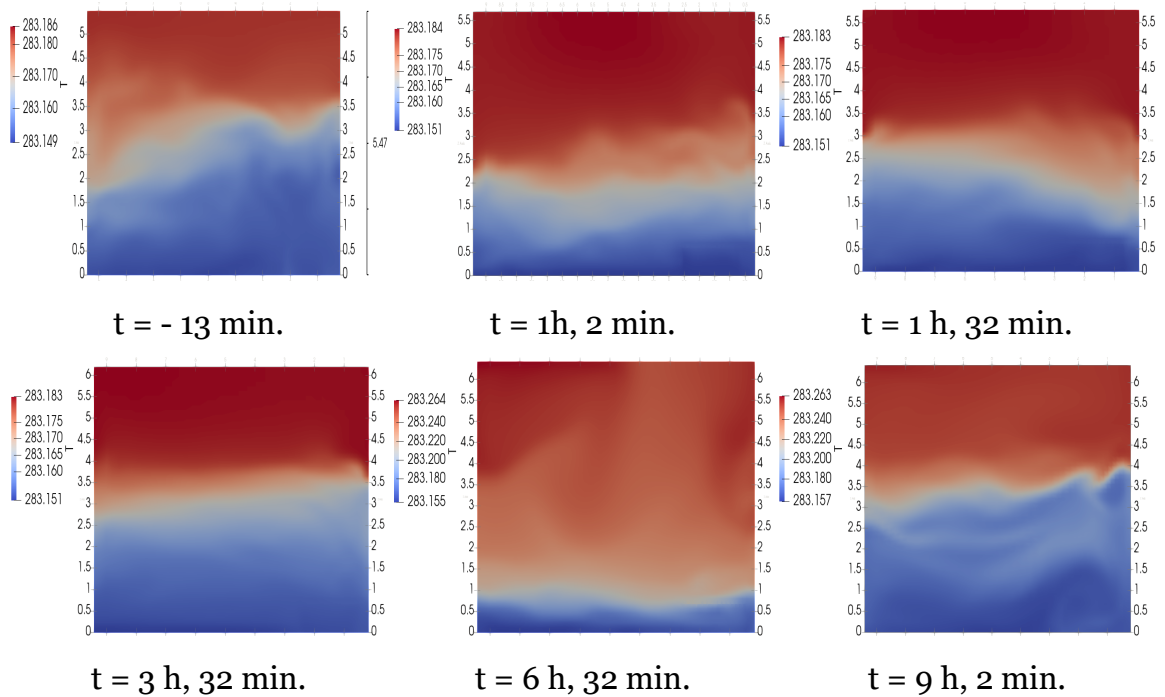
5th filling 27/9

Figure C.1.3: Simulated temperature distribution at the outlet at various times before, during and after the filling process the 27th September 2021.

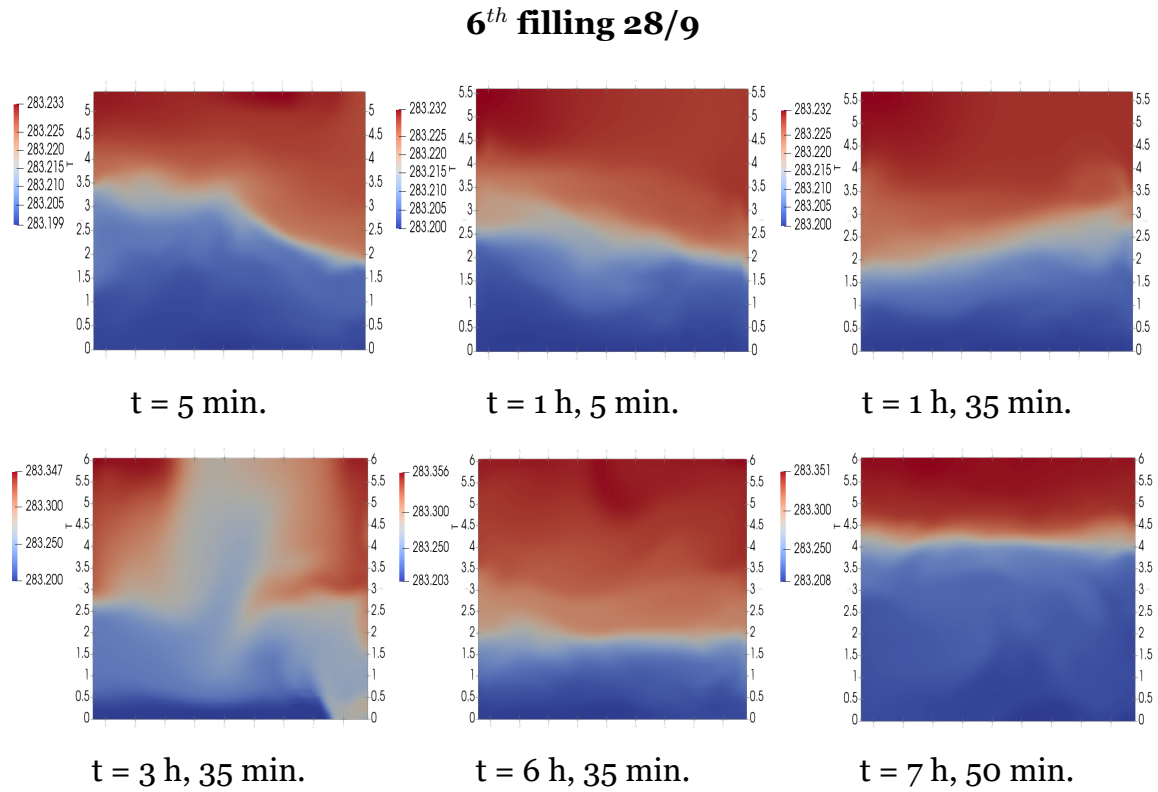


Figure C.1.4: Simulated temperature distribution at the outlet at various times before, during and after the filling process the 28th September 2021.

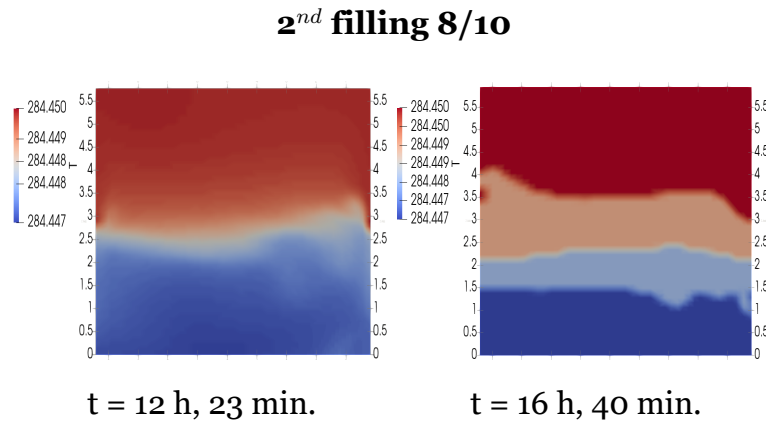


Figure C.1.5: Simulated temperature distribution at the outlet at various times before, during and after the filling process the 8th October 2021.

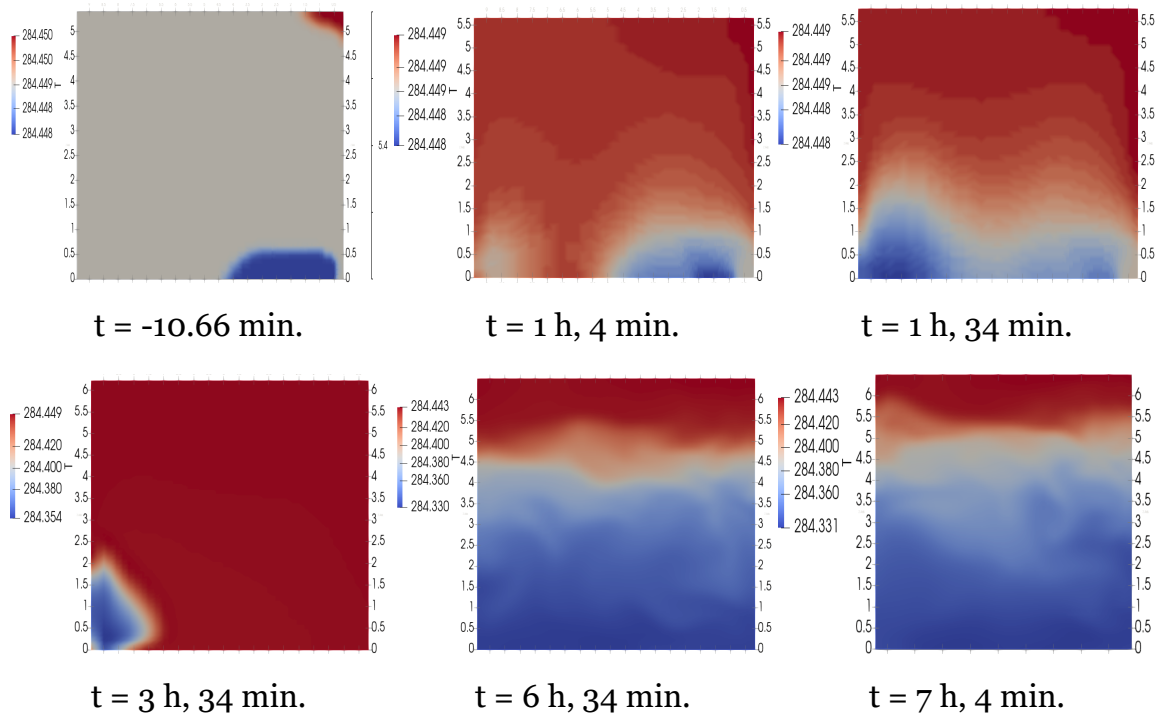
3rd filling 9/10

Figure C.1.6: Simulated temperature distribution at the outlet at various times before, during and after the filling process the 9th October 2021.

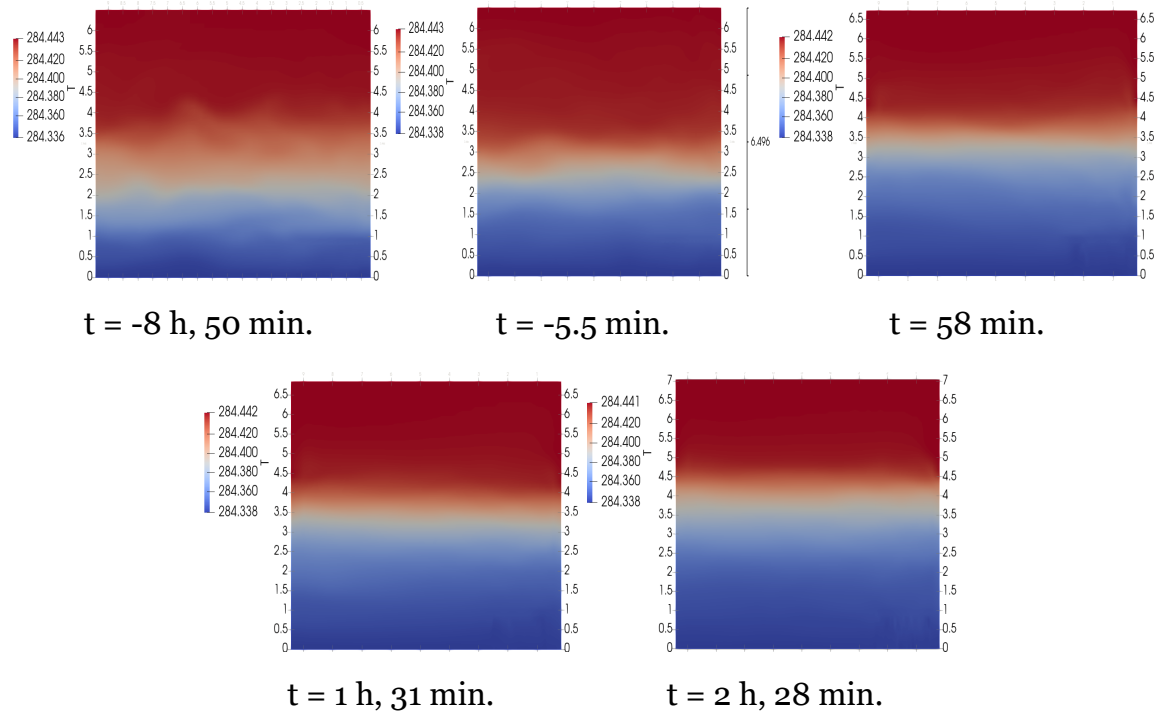
4th filling 10/10

Figure C.1.7: Simulated temperature distribution at the outlet at various times before, during and after the filling process the 10th October 2021.

C.1.1 Alternative simulations

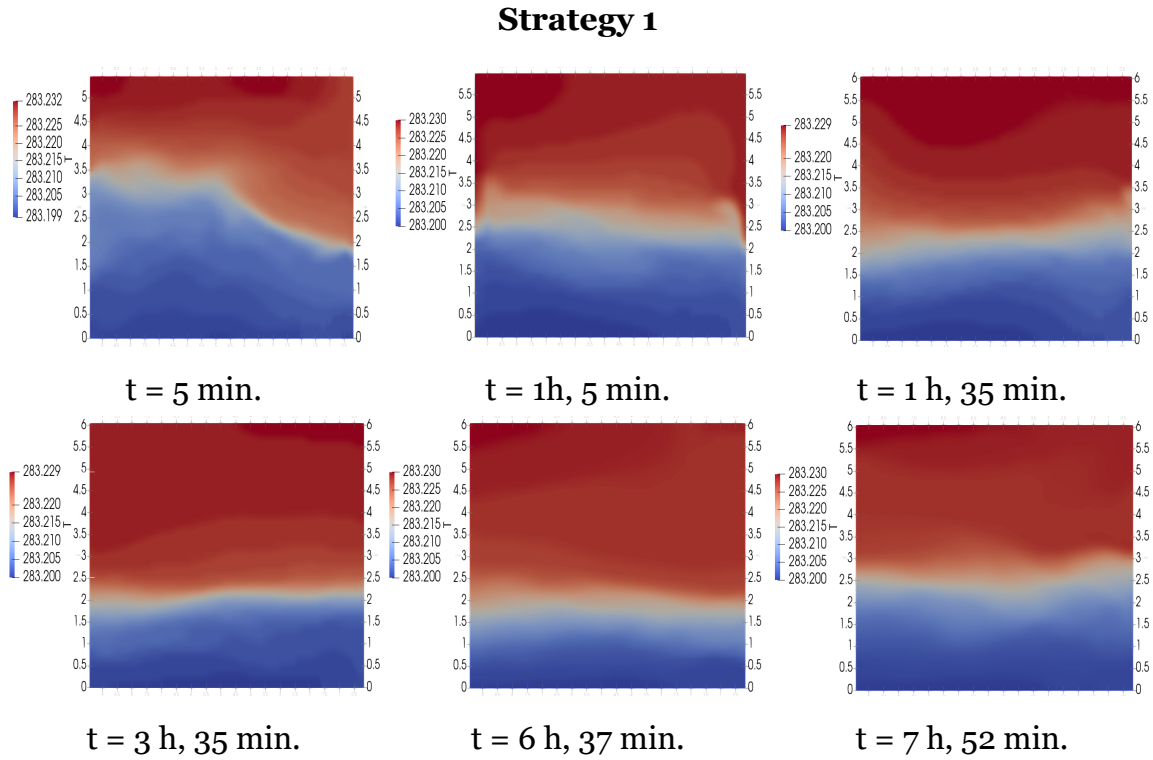


Figure C.1.8: Simulated temperature distribution at the outlet for the Strategy 1 filling process.

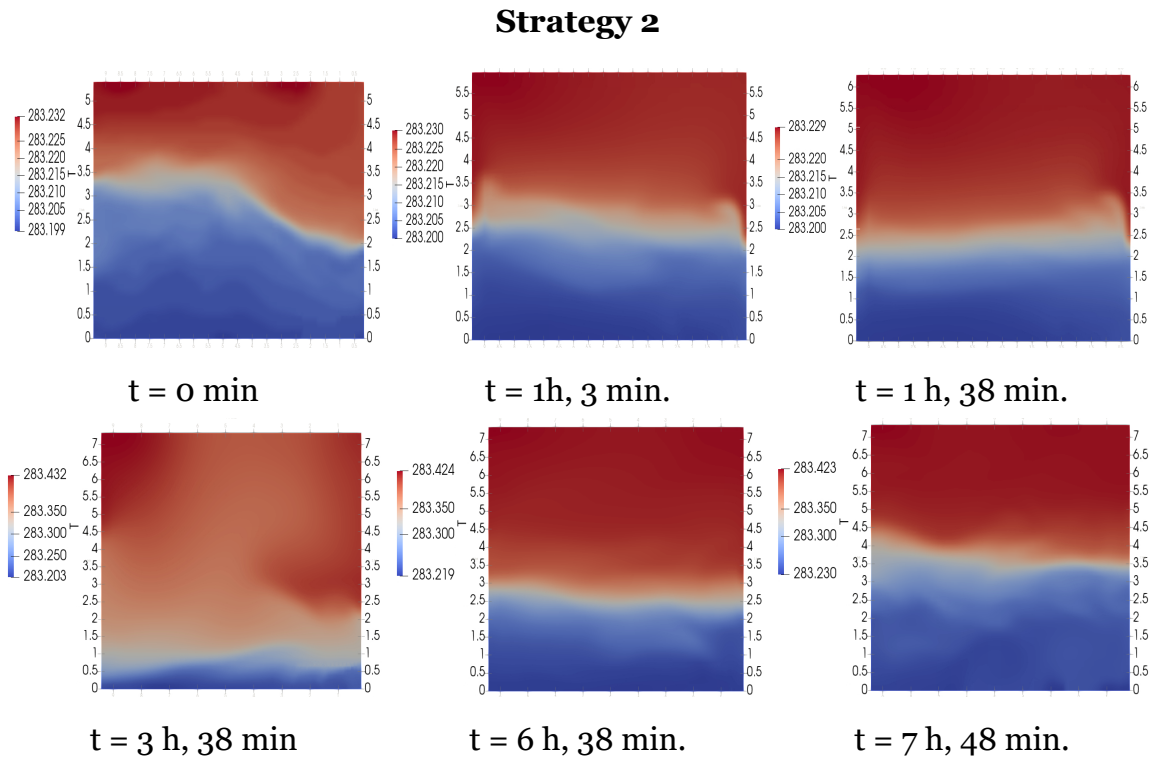


Figure C.1.9: Simulated temperature distribution at the outlet for the Strategy 2 filling process.

Strategy 3

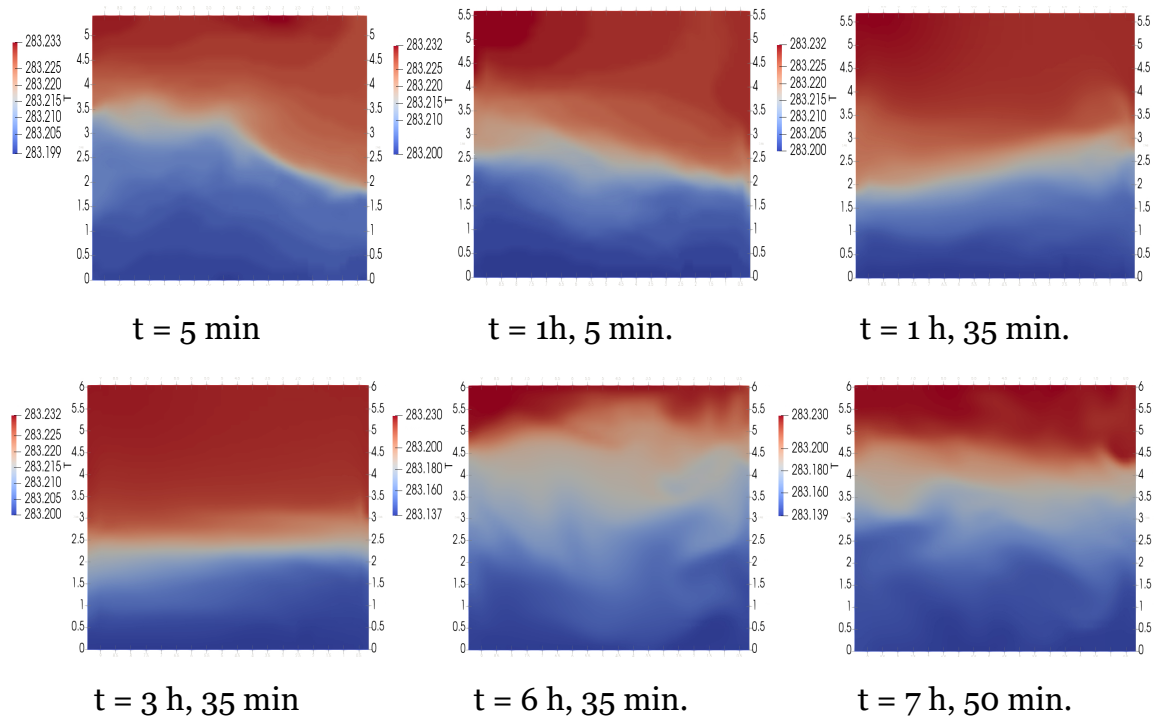


Figure C.1.10: Simulated temperature distribution at the outlet for the Strategy 3 filling process.

Strategy 4

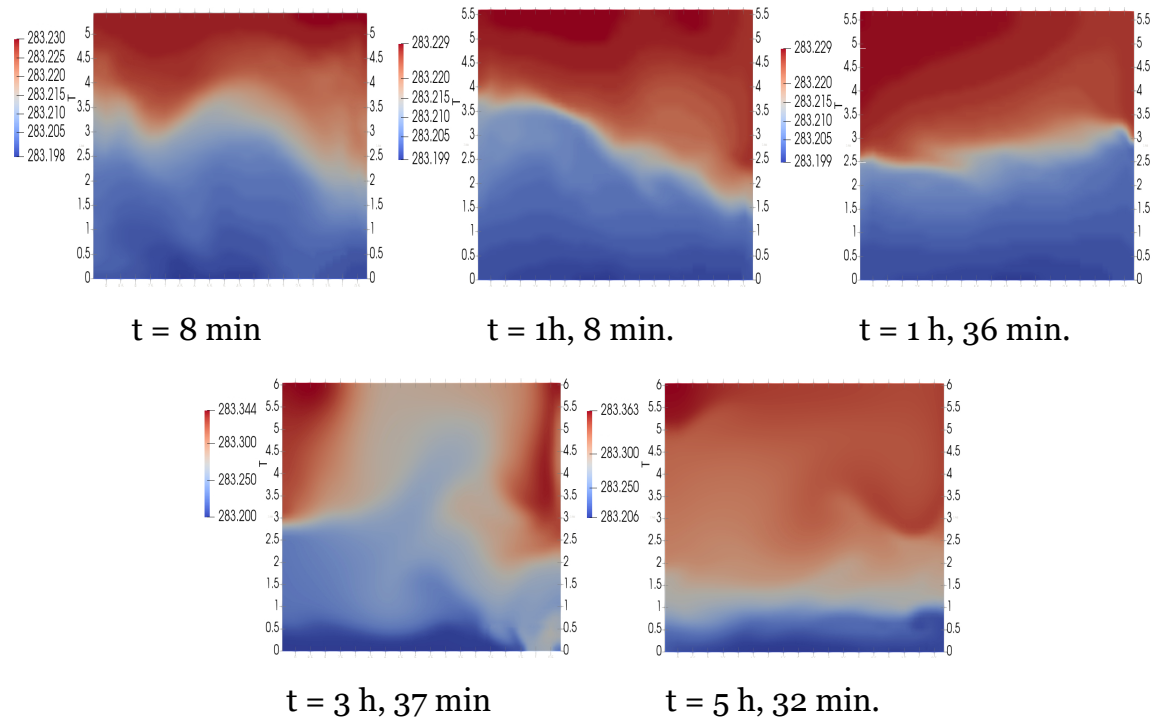


Figure C.1.11: Simulated temperature distribution at the outlet for the Strategy 4 filling process.

C.2 Flow cytometer data

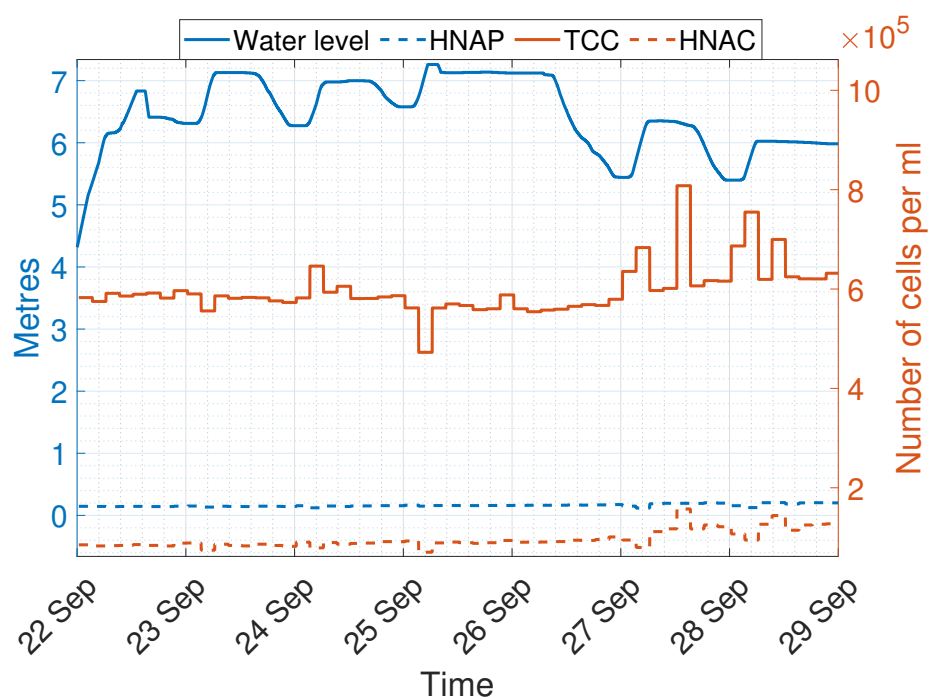


Figure C.2.1: Flow cytometer data for 22-28 September 2021.

