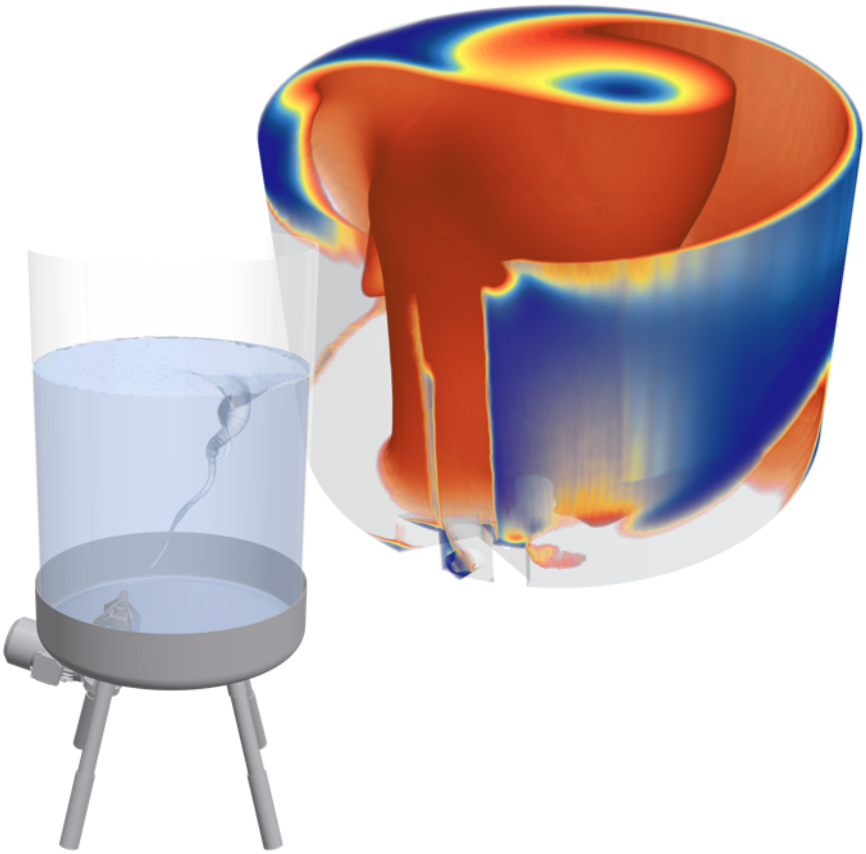
CHALMERS
UNIVERSITY OF TECHNOLOGY



Comparison of Commercial Finite Volume Method and Lattice Boltzmann Method Software for Mixing Applications

Master's thesis in Sustainable Energy Systems

HANNA LILJENZIN
ANGELIKI PEBES

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's Thesis in Sustainable Energy Systems

Comparison of commercial Finite Volume Method and Lattice Boltzmann Method software for mixing applications

HANNA LILJENZIN
ANGELIKI PEBES

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Fluid Dynamics
Chalmers University of Technology
Gothenburg, Sweden 2025

Comparison of commerical Finite Volume Method and Lattice Boltzmann Method software for mixing applications
HANNA LILJENZIN
ANGELIKI PEBES

Supervisor: Stewart Whitworth, FS Dynamics
Examiner: Lars Davidson, Department of Mechanics and Maritime Sciences

Master's Thesis 2025
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: CFD simulations of a biomedical tank (left) and an anaerobic digester (right), performed with the finite volume method as part of this thesis project.

Typeset in LaTeX
Gothenburg, Sweden 2025

Comparison of commerical Finite Volume Method and Lattice Boltzmann Method software for mixing applications

HANNA LILJENZIN
ANGELIKI PEBES
Department of Mechanics and Maritime Sciences
Division of Fluid Dynamics
Chalmers University of Technology

Abstract

This thesis investigates and compares two numerical methods for simulating industrial mixing processes: the Finite Volume Method, applied through the commercial software STAR-CCM+ executed on multiple CPU cores, and the Lattice Boltzmann Method, implemented in M-Star executed on a GPU. The methods were tested on three mixing cases: two small-scale biomedical tanks and a large-scale anaerobic digester. The comparison focuses on simulation accuracy, computational efficiency, and practical usability. Results show that both methods accurately capture important flow features in the biomedical tanks, with LBM offering significantly faster simulation times on GPU hardware. However, in the large-scale digester case, LBM faced challenges related to resolution, the need for local refinement, and limited flexibility in numerical models and solver settings. These factors contributed to long simulation times and less reliable results compared to the FVM approach. These findings suggest that while LBM is a promising method for smaller systems with low tank-to-impeller aspect ratios, while FVM remains more robust for complex, large-scale mixing applications.

Keywords: CFD, Mixing, Finite Volume Method, Lattice Boltzmann Method, Biomedical tank, Anaerobic Digester, GPU, CPU.

Preface

This report presents the outcome of our master's thesis project carried out at the Department of Mechanics and Maritime Sciences at Chalmers University of Technology during the spring of 2025. The thesis was performed at FS Dynamics with Stewart Withworth as supervisor. The project was examined by Lars Davidsson, head of the Division of Fluid Dynamics at Chalmers University of Technology.

Acknowledgements

We would like to express our gratitude to our supervisor Stewart Whitworth at FS Dynamics for his continuous support and valuable feedback throughout the thesis. His knowledge and guidance has been of great help during the entire project.

We would also like to thank Jorge Azevedo, our supervisor from FS Dynamics at the Lisbon office, for his insights and guidance regarding anaerobic digester tanks. His knowledge has been a great help during the simulation work. We are also very grateful for all the help from Johannes Wutz for his support and expertise related to the M-Star solver. His help has been essential for the Lattice Boltzmann Method simulation.

Furthermore, we would like to thank our examiner at Chalmers, Lars Davidson, for ensuring that the thesis process has been carried out smoothly and in accordance with the academic requirements.

Finally, we want to give out a huge thanks to all employees at FS Dynamics for welcoming us with warmth, sharing their knowledge, and contributing to a pleasant and supportive work environment during our time at the office.

Hanna Liljenzin, Angeliki Pebes, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BGK	Bhatnagar–Gross–Krook
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy
CPU	Central Processing Unit
CV	Coefficient of Variation
DES	Detached Eddy Simulation
FVM	Finite Volume Method
GCI	Grid Convergence Index
GPU	Graphics Processing Unit
ILES	Implicit Large Eddy Simulation
LBM	Lattice Boltzmann Method
LES	Large Eddy Simulation
MRF	Multiple Reference Frame
RANS	Reynolds-Averaged Navier–Stokes
RPM	Revolutions Per Minute
RSM	Reynolds Stress Model
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
SM	Sliding Mesh
VOF	Volume of Fluid

Contents

List of Figures

List of Tables

1

Introduction

Mixing processes are essential in many engineering applications, including biomedical systems and anaerobic digesters. A common approach to studying these processes is Computational Fluid Dynamics (CFD) simulations. In large-scale digesters, experimental measurements are often impractical due to limited accessibility and complex flow behavior. In contrast, biomedical tanks typically involve smaller domains where more detailed flow structures or particle distributions can be of interest. This level of detail significantly increases the computational cost, even for relatively compact geometries. To make simulations more efficient, alternative methods that reduce computational time but still capture important flow physics are of interest. This study investigates two different numerical approaches to simulate industrial mixing processes. This chapter describes the background of the project along with previous research followed by the purpose, goals, and demarcations of the study.

1.1 Background

The evaluation of mixing characteristics is fundamental in many industrial applications. CFD can be a valuable tool for gaining deeper insight into key parameters and flow patterns, as it enables detailed analysis without the need for physical experiments. The Finite Volume Method (FVM) has been an industry standard in CFD for several decades and is widely implemented in both commercial and academic software. The Lattice Boltzmann Method (LBM) has emerged as a promising alternative to conventional CFD methods such as the Finite Volume Method (FVM), since it offers natural parallelization and straightforward implementation of non-equilibrium physics [3]. This makes the method especially suitable for large-scale or multiphysics problems where conventional methods either become too expensive or rely on simplifying assumptions. The performance is studied using both numerical approaches in two separate software: Siemens Simcenter STAR-CCM+ (hereafter referred to as STAR-CCM+), which is based on the FVM, and M-Star CFD, which employs an LBM-based solver [4, 5].

This thesis investigates and compares the performance of FVM and LBM in mixing simulations using two different types of tanks: two small-scale biomedical tank and a large-scale digester tank. The tank types will be used for three different simulations cases presented below:

Case 1: Biomedical tank

Mixing is performed in a stirred tank with the impeller positioned off-center. The objective is to validate the flow characteristics, with the power number and pumping number as key parameters. The vortex formation and the mixing time will also be assessed.

Case 2: Digester tank

Mixing in digester tank where the mixing is driven by two impellers combined with a draft tube. The fluid in the tank is non-Newtonian and shear thinning. The main objective is to evaluate the mixing time and flow field.

Case 3: Biomedical tank with particles

Mixing is driven by a pitched blade impeller. The objective is to validate the flow field and distribution of particles injected into the tank.

1.2 Previous Research

A literature study was carried out to gain insight into previous work on CFD simulations of industrial mixing, particularly in stirred tanks and agitated digester tanks. A brief overview of relevant studies is presented below.

Larsson and Lindenbaum [2] conducted experimental measurements of mixing performance in a stirred tank, which was later used for corresponding CFD simulations. A vortex formed in the tank under operational conditions, and their work focused on evaluating which turbulence models could accurately capture the vortex behavior. By using the same geometry in both experiments and simulations, they enabled a direct comparison of power number, pumping number, and mixing time. Their results showed that the Reynold Stress Model (RSM) was sufficiently accurate for representing the vortex dynamics in the CFD simulations.

Shan et al. [1] performed experiments in an unbaffled stirred tank with a pitched-blade turbine to measure local solids concentrations using a fiber-optic probe. The same setup was later used for CFD simulations of the liquid-solid flow, employing an Eulerian multiphase approach combined with the standard $k - \varepsilon$ turbulence model. Their study showed good agreement between experimental and simulated solid holdup distributions. To capture the rotating impeller and resulting central vortex, they introduced a "vector distance" method that allowed them to approximate the impeller geometry within a fixed grid, enabling simulation of complex flow features such as the vortex near the shaft.

Building on Shan et al.'s work, Pukkella et al.[6] developed and validated a three-step CFD framework for solid–liquid mixing in vortex-forming stirred tanks. They first used a steady-state VOF simulation to capture the vortex shape, which was then used to refine the liquid domain. In the final step, Eulerian multiphase simulations

were conducted using the realizable $k - \varepsilon$ model and Gidaspow drag law to predict the solid distribution. The model reproduced the experimentally measured solid volume fractions at various heights with good agreement.

Previous studies on anaerobic digesters (AD) have commonly treated sludge as a non-Newtonian fluid, typically described using the power-law model [7, 8]. This model captures the shear-thinning behavior by allowing viscosity to vary with shear rate, within a specific range. Outside this range, viscosity is capped at a maximum or minimum value. Alternatively, the Herschel–Bulkley model has been employed to account for yield stress and shear-dependent viscosity, offering a more comprehensive description for certain systems [9, 10].

While AD systems inherently include three phases: liquid, solid, and gas, most CFD studies simplify the problem by considering single- or two-phase flow [11, 8]. Modeling all three phases introduces significant complexity in terms of required input data and computational cost, due to the need for detailed interphase interactions and boundary conditions.

In terms of turbulence modeling, RANS approaches, particularly the k–ε family, are widely used due to their relatively low computational cost [12, 13, 8]. A previous study evaluated six different RANS turbulence models , concluding that the standard k–ε model is more suitable for mechanically agitated digestive systems than the alternatives [12].

For simulating mechanical mixing in stirred digesters, techniques like the Moving Reference Frame (MRF) and Sliding Mesh (SM) have been frequently applied [14, 15, 16]. These methods divide the domain into rotating and stationary zones to account for impeller motion, while in simpler cases, a Single Reference Frame (SRF) may be sufficient [17].

1.3 Purpose and Objective

The purpose of this thesis is to evaluate and compare the performance of two numerical methods, FVM and LBM, by simulating mixing processes in fluid systems of different scales and complexities. This will be conducted by applying both methods to representative cases of two biomedical tanks and a digester tank. The study aims to compare the methods and validate the results with experimental data if available. The comparison between the two software tools will focus on aspects such as:

- Evaluate mixing parameters such as power number, particle distribution, and mixing time.
- Assess and compare simulation accuracy, computational time and model setup.
- Identify strengths and limitations of each method in relation to geometry size, tank aspect ratios, and fluid rheology.

1.4 Demarcations

This master thesis project will have some demarcations, mostly in order to keep the project within reasonable scope for the time limit. The demarcations determined for this thesis work are the following:

- No comparison between different turbulence models for FVM simulations will be conducted. A suitable turbulence model for each case will be chosen by reviewing literature and research done on similar setups.
- The evaluation of the numerical methods will be limited to comparing two types of tanks for mixing applications, which may not offer a comprehensive basis for assessing broader applications.
- Each of the two numerical methods will be evaluated using a single software, which may introduce limitations related to the specific implementation rather than the numerical methods themselves.

2

Theory

This chapter presents the theoretical background relevant to the thesis, focusing on aspects that are essential for the applied methodology. The chapter begins with an overview of industrial mixing, followed by a brief introduction to the numerical methods used. Next, all relevant concepts related to CFD are described, and finally, the software and hardware used in the project are presented.

2.1 Industrial Mixing

Mixing is a key operation in a wide range of industrial processes and is essential for ensuring product quality and process efficiency [18]. Although much of the foundational understanding of mixing originates from the chemical industry, it is also widely applied in sectors such as food, pharmaceuticals, paper, plastics, ceramics, and rubber. In these industries, mixing plays a central role in combining materials, dispersing phases, and facilitating various process steps.

This thesis investigates different types of industrial mixing. This section presents the underlying physical principles and dimensionless numbers relevant to the different mixing configurations analyzed in the study.

2.1.1 Anaerobic Digestive Mixing

Anaerobic digestion is a biological process for treating organic wastes, diminishing excess biological sludge from wastewater treatment plants and recovering energy from waste products [19]. Currently, three types of mixing are commonly used in industrial anaerobic digesters: mechanical mixing, gas mixing, and jet mixing. Fast-rotating impellers with small diameters, often combined with a draft tube, are also employed in these mixing processes. The impeller drives the flow through the draft tube between the top and bottom of the tank, thereby enhancing overall mixing efficiency [20]. A key factor influencing the mixing process is the rheological behavior of the input material [19]. These input materials, such as wastewater and slurries from food processing plants, are typically non-Newtonian fluids.

2.1.2 Non-Newtonian Fluid

Non-Newtonian fluids do not exhibit the same linear relation between shear stress and shear rate as Newtonian fluids, where the shear stress is directly proportional to the shear rate. While Newtonian fluids generally have constant viscosity, non-Newtonian fluids experience fluctuating viscosity. Non-Newtonian fluids in mixing applications can lead to non-uniform and non-ideal flow conditions, resulting in incomplete mixing and stagnant zones, which significantly affect the efficiency of the process. For non-Newtonian fluids the viscosity depends on the shear rate magnitude $\dot{\gamma}$ by following a power-law relationship [21]:

$$\mu = K|\dot{\gamma}|^{n-1}, \tag{2.1}$$

where K $[Pa \cdot s^n]$ is the consistency coefficient and n is the power law index. When n is smaller than 1 the fluid is shear-thinning. A non-Newtonian fluid can also be described by the Herschel-Bulkley model in which flow occurs only if the shear stress exceeds a critical value τ_0 according to a power law:

$$\mu = \tau_0|\dot{\gamma}|^{-1} + K|\dot{\gamma}|^{n-1}. \tag{2.2}$$

2.1.3 Reynolds Number

The Reynolds number in a stirred tank is defined as

$$Re = \frac{\rho N D^2}{\mu}, \tag{2.3}$$

where D is the impeller diameter, N is the rotational speed, ρ is the density of the fluid being mixed and μ is the dynamic viscosity [22].

2.1.4 Power Number

The power number is a non-dimensional measure of the power exerted on the fluid by the impeller. The power number can be determined by using the torque applied to the impeller [23]. Then the power number, $N_{P\tau}$ is defined as

$$N_{P\tau} = \frac{P}{\rho N^3 D^5} \tag{2.4}$$

where P is the power defined as:

$$P = 2\pi N\tau \tag{2.5}$$

and τ is the torque applied on the impeller.

2.1.5 Pumping number

Pumping is the amount of material discharged by the rotating impeller. The pumping number, N_Q is defined as

$$N_Q = \frac{Q}{ND^3} \tag{2.6}$$

where Q is the volumetric flow rate exiting a cylindrical surface enclosing the impeller [24].

2.1.6 Mixing Time

There are several ways to determine the mixing time in an agitated tank. Mixing time refers to the duration required for the contents of a tank to reach a certain degree of homogeneity, often evaluated through the concentration of a tracer [25].

A commonly used criterion is 95% homogeneity, with the corresponding mixing time denoted as τ_{95}. This can be assessed either locally or globally. Local mixing time is determined by observing the concentration at a specific location in the tank. When the normalized concentration at that point falls within the range $0.95 \cdot \frac{c_t}{\overline{c_t}}$ to $1.05 \cdot \frac{c_t}{\overline{c_t}}$, where c_t is the concentration at a specific point and $\overline{c_t}$ is the average concentration in a fully mixed state. The time at which this condition is first met is defined as τ_{95}.

Another way to determine the mixing time is by the coefficient of variation (CV). The degree of mixing is quantified using the spatial variance of the tracer concentration and is defined as follows:

$$\mathrm{CV} = \sqrt{\frac{\sum_i \left(\frac{c_i - \bar{c}}{\bar{c}}\right)^2 \Delta V_i}{\sum_i \Delta V_i}} \tag{2.7}$$

where i denotes the index of a monitoring point with volume ΔV_i [25]. The threshold for CV is 0.0283 for 95% homogeneity.

2.2 Finite Volume Method

FVM is currently the most widely adopted numerical method for performing CFD simulations. In FVM simulations, the flow field is resolved by solving the continuity and momentum conservation equations, commonly known as the Navier-Stokes equations [3]. FVM is based on the principle of local conservation, making it well-suited for CFD calculations [26]. To solve the equations numerically, the computational domain is divided into discrete cells, where the Navier-stokes equations are reformulated into a system of linear algebraic equations for each cell. These equations are solved iteratively with a given set of boundary conditions [3].

The Navier-stokes equations are expressed as:

$$\nabla \cdot U = 0 \tag{2.8}$$

$$\rho\frac{\partial U}{\partial t} + \rho U \cdot \nabla U = -\nabla p + \mu\nabla^2 U \tag{2.9}$$

where U is the velocity vector, ρ is the fluid density, p is the pressure and μ is the dynamic viscosity [3].

In the discretization process, the conservation laws are applied in the discrete cells [27]. The conservation laws are enforced over each cell to compute the values of the flow variables at discrete points called nodes. The number of nodes in the domain depends on the type of cells used, which can be for example triangular or quadrilateral. The derivation of the finite volume formulation is carried out starting from a transport equation involving convection and diffusion processes. The general form of such a transport equation is expressed as:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\mathbf{u}\phi) = \nabla \cdot (D\nabla\phi) + S_\phi \tag{2.10}$$

where ρ is the density, ϕ is the conservative form of all fluid flow, u is the velocity, D is the diffusion coefficient and S_ϕ is the source term. When integrating this equation over a given domain V, equation 2.11 is obtained,

$$\int_V \frac{\partial(\rho\phi)}{\partial t}dV + \int_V \nabla \cdot (\rho\mathbf{u}\phi)dV = \int_V \nabla \cdot (D\nabla\phi)dV + \int_V S_\phi dV. \tag{2.11}$$

The terms $\frac{\partial}{\partial t}$ and $\int_V(\rho\phi)dV$ commute each other. For a fixed domain, the equation reduces to:

$$\int_V \frac{\partial(\rho\phi)}{\partial t}dV = \frac{\partial}{\partial t}\int_V \rho.\phi dV. \tag{2.12}$$

Using the Gauss-divergence theorem, the volume integral can be transferred into surface integrals

$$\int_V \nabla \cdot (\rho\mathbf{u}\phi)dV = \int_S \mathbf{n} \cdot (\rho\mathbf{u}\phi)dS \tag{2.13}$$

$$\int_V \nabla \cdot (D\nabla\phi)dV = \int_S \mathbf{n} \cdot (D\nabla\phi)dS \tag{2.14}$$

where n is the normal to the surface S. If equations 2.12-2.14 are substituted into equation 2.11, the required integral form for the FVM (to solve the conservation equations in the nodes) is obtained as

$$\frac{\partial}{\partial t}\int_V \rho\phi dV + \int_S \mathbf{n} \cdot (\rho\mathbf{u}\phi)dS = \int_S \mathbf{n} \cdot (D\nabla\phi)dS + \int_V S_\phi dV. \tag{2.15}$$

The conservation laws are then expressed in their integral form and applied to each cell individually. The resulting surface and volume integrals are approximated using quadrature formulas, which estimate the integrals based on the values at the computational nodes. These expressions are then assembled into a system of algebraic equations, one equation per cell. Finally, this system is solved numerically to determine the unknown values of the physical variables at the computational nodes. This approach accounts for the streams and collisions of individual molecules in the fluid, providing a detailed description of intermolecular interactions [3].

2.3 Lattice Boltzmann Method

In contrast to FVM, LBM is a numerical method that does not directly solve the Navier-Stokes equations [3]. While the Navier-Stokes-based solvers rely on discretizing the governing differential equations with a given set of boundary conditions on a computational grid network, the lattice Boltzmann equations are solved by satisfying the governing equations in mesoscopic scale using distribution functions with streaming and collision processes. The LBM is an inherently time-dependent method [28]. It is possible to obtain steady solutions, but it usually requires longer simulation times, since a larger number of iterations are required compared to methods tailored for steady problems. The governing equations and relevant theoretical background related to the LBM implementation used in this thesis are presented in the following section.

2.3.1 The Lattice Boltzmann Equations

In LBM, continuum phenomena such as fluid mechanics and thermodynamics are described at the macroscopic level, while the kinetic theory treats the domain at a microscopic level, modeling it as a system of discrete particles [3]. However, simulating individual atoms or molecules would require an immense amount of computational time due to the large number of particles involved. To address this, LBM reduces the number of simulated particles by considering groups of atoms or molecules instead of individual atoms or molecules. This changes the simulation scale from microscopic to mesoscopic. Additionally, the degrees of freedom are limited by constraining particle movement to a finite set of discrete directions.

In LBM, a set of fictitious particles stream and collide within a discrete lattice domain [3]. The distribution function, f(r,c,t), describes the probability of finding particles with a velocity in the range c to c+dc and a position within r to r+dr at a given time t. This eliminates the need to track individual particles, as required in molecular dynamics, thereby significantly reducing computational cost and making LBM well-suited for large domains. The transition between initial and final states is governed by the collision operator, Ω, and the equation of particles numbers can be written as:

$$f\left(r + cdt, c + \frac{F}{m}dt, t + dt\right) drdc - f(r, c, t)drdc = \Omega(f)drdcdt. \tag{2.16}$$

By limiting dt→0:

$$\frac{df}{dt} = \Omega(f). \tag{2.17}$$

As the distribution function f is a function of c, r and t, the Boltzmann transport equation can be written as below:

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial \vec{r}} + \frac{F}{m}\frac{\partial f}{\partial c} = \Omega. \tag{2.18}$$

Macroscopic fluid quantities such as velocity and pressure are captured by modeling the movements of fluid particles in a discretized domain with uniform cartesian cells, where each cell holds several distribution functions. The movement of fluid particles is restricted to discrete directions, with the distribution functions representing the number of particles traveling along each direction. The number of distribution functions depend on the specific method used. These methods employ a structured lattice arrangement, denoted as DnQm, where n represents the spatial dimension and m indicate the total number of discrete velocity directions available for particle streaming.

2.3.2 Bhatnagar-Gross-Krook Approximation

To solve the Boltzmann equation, the collision operator Ω, that is a function of f, must be given. By using the Bhatnagar-Gross-Krook (BGK) approximation, offering a simple model for the collision operator, the Boltzmann equation without external forces can be described by,

$$\frac{\partial f}{\partial t} + \mathbf{c} \cdot \nabla f = \frac{1}{\tau}(f^{eq} - f) = \omega(f^{eq} - f) \tag{2.19}$$

where $\omega = \frac{1}{\tau}$ is the collision frequency with τ as the relaxation factor, and f^{eq} represents the local equilibrium distribution function. When discretizing the Lattice Boltzmann equation in space x and time t, and incorporating external forces, the equation in direction k takes the following form,

$$f_k(\mathbf{x} + \mathbf{c}_k\Delta t, t + \Delta t) = f_k(\mathbf{x}, t) + \frac{\Delta t}{\tau}\left[f_k^{eq}(\mathbf{x}, t) - f_k(\mathbf{x}, t)\right] + \Delta t F_k \tag{2.20}$$

where $f_k^{eq}(\mathbf{x}, t)$ can be described as

$$f_k^{eq}(\mathbf{x}, t) = \omega_k \rho \left[1 + \frac{\mathbf{c}_k \cdot \mathbf{U}}{c_s^2} + 0.5\left(\frac{(\mathbf{c}_k \cdot \mathbf{U})^2}{c_s^4} - \frac{\mathbf{U} \cdot \mathbf{U}}{c_s^2}\right)\right], \tag{2.21}$$

where c_s^2 is

$$c_s^2 = \left(\frac{\partial p}{\partial \rho}\right)_s . \tag{2.22}$$

In these equations, Δt is the lattice time step, c_s is the speed of sound, c_k is the discrete lattice velocity, F_k the external force and r the lattice fluid density. ω_k is a weighing factor that varies for different DnQm. The domain is typically discretized such that $c_k \Delta t$ equals the distance between two neighboring lattice nodes. As a result, the distribution function $f_k(x,t)$ moves to the next lattice node in the direction of the discrete lattice velocity c_k after one time step.

2.3.3 Chapman-Enskog Expansion

The Navier-Stokes equations describe the dynamics of a fluid and through the Chapman-Enskog expansion, these equations can be derived from the Lattice Boltzmann equations [29]. In the derivation, one can consider a D2Q9 model with the following multiscale expansions:

$$f_i = f_i^{(0)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} + \cdots \tag{2.23}$$

where ε is a small number proportional to the Knudsen number. The second-order Taylor series expansion of the lattice BGK equation yields:

$$D_i f_i + \frac{\delta_t}{2} D_i^2 f_i = -\frac{1}{\tau \delta_t}(f_i - f_i^{eq}) + O(\delta_t^2) \tag{2.24}$$

$$D_i = \partial_t + \mathbf{c}_i \cdot \nabla = \partial_t + c_{i\alpha}\partial_\alpha \tag{2.25}$$

Substituting equation 2.23 into equation 2.24 and collecting terms of the same order in ε, the first-order equation can be obtained as:

$$D_i^{(0)} f_i^{(0)} = -\frac{1}{\tau \delta_t} f_i^{(1)} \tag{2.26}$$

To derive the macroscopic equations at order ε, equation 2.26 is first summed over all directions i to obtain the conservation of mass. Then, by multiplying equation 2.26 by the discrete velocity vector $\mathbf{c}_i$ and summing over i, the momentum conservation equation is recovered. These steps correspond to taking the zeroth and first moments of the equation, respectively:

$$\partial_{t_0} \rho + \nabla_0 \cdot (\rho \mathbf{u}) = 0, \tag{2.27}$$

$$\partial_{t_0}(\rho \mathbf{u}) + \nabla_0 \cdot \boldsymbol{\pi}^{(0)} = 0, \tag{2.28}$$

where $\boldsymbol{\pi}^{(0)} = \sum_i \mathbf{c}_i \mathbf{c}_i f_i^{(0)}$ is the zeroth-order momentum flux tensor.

To recover the Navier-Stokes equations, the next step involves incorporating the first-order correction term $f_i^{(1)}$ into the momentum flux [29]. This introduces viscous effects. By expressing the total flux tensor as $\boldsymbol{\pi} = \boldsymbol{\pi}^{(0)} + \varepsilon \boldsymbol{\pi}^{(1)}$ and applying the Chapman Enskog expansion to second order in ε, the incompressible Navier-Stokes equations are recovered:

$$\nabla \cdot \mathbf{u} = 0, \tag{2.29}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}. \tag{2.30}$$

Here, the kinematic viscosity ν is related to the LBM relaxation time τ through:

$$\nu = c_s^2 \delta_t \left(\tau - \frac{1}{2} \right) \tag{2.31}$$

Thus, through the Chapman-Enskog expansion, the LBM is shown to be consistent with the Navier-Stokes equations in the incompressible limit.

2.3.4 No-slip Boundary Condition

Implementing boundary conditions is less straightforward in LBM compared to other conventional methods [3]. The no-slip condition is commonly implemented using the bounce-back condition. In this approach, every distribution function at the boundary is replaced by the distribution function in opposite direction, assuming that the incoming and outgoing distribution functions are equal in magnitude but opposite in direction. During the collision, equation 2.32 is applied to implement the no-slip boundary condition,

$$f_p^{\text{in}}(x, t+1) = f_q^{\text{out}}(x, t), \tag{2.32}$$

where p and q represent opposite directions.

2.4 Computational Fluid Dynamics

In the following section various concepts and modeling strategies in CFD is introduced. Unless otherwise specified, the concepts discussed apply to CFD using both FVM and LBM.

2.4.1 Moving Bodies

In CFD there exists several methods for simulating moving bodies within a flow field. These approaches differ in how the motion is represented and how the interaction between the fluid and the moving geometry is handled. The following section will present the methods used in this thesis project.

2.4.1.1 Multiple Reference Frame

The MRF model is a steady state method used in FVM to simulate rotating equipment such as impellers [22]. Stationary parts like tank walls and baffles are modeled in a fixed frame, while the impeller is placed in a rotating frame where it appears stationary. For accurate results, the mesh around the rotating zone must be symmetric about the axis of rotation. The flow is solved relative to each frame, and information is exchanged across the interface. Although the impeller does not physically rotate, its effects on the surrounding flow are captured through the rotating frame.

2.4.1.2 Sliding Mesh

The SM model is a transient method in FVM for simulating rotating flows [30]. Unlike MRF, it physically rotates the mesh around the impeller during the simulation, allowing accurate capture of the full transient startup. The domain is split into two zones: an inner cylindrical zone that contains the rotating part and an outer stationary zone. These zones slide relative to each other along a defined interface, where the inner mesh rotates a fixed angle each time step.

2.4.1.3 Immersed Boundary

Conventional methods for handling moving solid bodies involves generating a mesh that conforms to the exact shape of the object [31]. These type of methods first defines a surface grid covering the boundary, Γ_b, and then generates the mesh in the regions occupied by the fluid, Ω_f, and the solid, Ω_s. On the other hand, when using the immersed boundary method, Γ_b still covers the body. Then a cartesian grid covering the fluid domain is generated with no regard to Γ_b. The solid now cuts through the cartesian grid and a boundary condition is needed to modify the equations at the boundary. In order to account for the boundary conditions on the immersed boundary, there are two different approaches. Both approaches use source terms in the governing equations in order to reproduce the effect of a boundary, which is incorporated before discretization in the first approach. In the second approach, the source term is incorporated after the discretization.

2.4.2 Courant Number

The Courant number, C, is a dimensionless number defined as:

$$C = \frac{u\Delta t}{\Delta x}, \tag{2.33}$$

where u is the velocity magnitude, Δt is the time step and Δx is the length interval. The Courant Number is commonly used to determine a suitable time step size [26]. The time step size is limited by the requirement that the time step should be shorter than the time that is taken to transport past the cell.

2.4.3 Wall Functions

At a solid wall the no-slip condition applies, meaning that the relative velocity between the fluid and the wall is zero [26]. When a flow is constrained by a solid wall, the inertial and viscous forces change going from a layer near the wall to a region further from the wall, creating a boundary layer. The boundary layer has a thickness, δ, which is often determined by the distance from the wall, y, where the fluid reaches 99% of the free-stream velocity. The inner region of the boundary layer, $0 < y < 0.2\delta$, is usually divided into sub-layers, defined by the dimensionless wall distance $y+$.

$$y+ = \frac{yu^*}{\nu} \tag{2.34}$$

where u^* is the wall friction velocity and ν is the kinematic viscosity. The innermost region, where $0 < y^+ < 5$, is referred to as the viscous sub layer. In this layer, the flow is nearly laminar and dominated by viscous stresses. Further from the wall, the buffer layer is defined by $5 < y^+ < 30$, where viscous and turbulent stresses are of comparable magnitude. In the outer region, the log layer is defined by $30 < y^+ < 400$, where turbulent stresses dominate the flow behavior.

In this near wall region, particular turbulence models are not valid. Rapid variations of the flow variables occur, implying the requirement of a very fine computational mesh to resolve the large gradients of flow variables [26]. To model the near-wall region, wall functions can be applied. The core idea is to apply boundary conditions at a certain distance from the wall, thereby avoiding the need to solve the turbulence model in the immediate near-wall region. In this way, wall functions account for the rapid variations in flow variables near the wall without requiring the viscous sub layer to be fully resolved. The use of wall functions requires that the first grid point adjacent to the wall lies within the log layer of the boundary layer. Ideally, it should be placed near the lower limit of this region to maximize the number of grid points captured within the boundary layer.

2.4.4 Turbulence Models

Turbulence is encountered in most flows in nature and in industrial applications, since turbulence enhances heat- and mass-transfer rates [26]. Therefore, turbulence modeling becomes one of the key elements in CFD. The turbulence models relevant for this thesis will be presented in this section.

2.4.4.1 Reynolds Averaged Navier-Stokes Equations

The instantaneous variables in the Navier-Stokes equations (equations 2.8 and 2.9) can be split into a mean part and a fluctuating part [26]. This is called Reynolds decomposition and is denoted as follows:

$$U_i = \langle U_i \rangle + u_i \tag{2.35}$$

and

$$P = \langle P \rangle + p. \tag{2.36}$$

By substituting the instantaneous variables with equation 2.35 and 2.36 into equation 2.8 and 2.9, the general form of the Reynolds Averaged Navier-Stokes (RANS) equations can be obtained.

$$\frac{\partial \langle U_i \rangle}{\partial t} + \langle U_j \rangle \frac{\partial \langle U_i \rangle}{\partial x_j} = -\frac{1}{\rho}\frac{\partial}{\partial x_j}\left[\langle P \rangle \delta_{ij} + \mu\left(\frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i}\right) - \rho \langle u_i u_j \rangle\right] \tag{2.37}$$

There are multiple turbulence models based on the Reynolds decomposition, since the flow then is described statistically by the mean flow velocity and turbulence quantities [26]. By averaging over a suitable time interval, the turbulent fluctuations are separated from the non-turbulence quantities. The resulting set of equations is commonly referred to as the RANS equations.

The term $\rho \langle u_i u_j \rangle$ is known as the Reynolds stresses. To close equation 2.37, this term must be modeled. One common approach is to relate the Reynolds stress tensor to the mean velocity field. This is known as the Boussinesq approximation, which assumes that the Reynolds stresses are proportional to the mean velocity gradients. It treats the transport of momentum by turbulence as a diffusive process and introduces a turbulent viscosity. The Boussinesq approximation reads

$$\frac{\tau_{ij}}{\rho} = -\langle u_i u_j \rangle = \nu_T\left(\frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i}\right) - \frac{2}{3}k\delta_{ij}, \tag{2.38}$$

where ν_T is the turbulent viscosity [26]. The second term on the right-hand side accounts for the normal stresses. Different turbulence models based on the RANS equations and the Boussinesq approximation use various methods to determine this viscosity. These models are typically categorized by the number of additional transport equations they require to describe the local turbulence characteristics. Zero-equation models use no additional PDEs, while one- and two-equation models solve one or two transport equations alongside the RANS equations.

2.4.4.2 $k-\varepsilon$ Model

The most widely used turbulence model in CFD is the $k-\varepsilon$ model [26]. The equations for the $k-\varepsilon$ model are simplifications of the exact transport equations for k and ε, which makes the model one of several possible closures by which the RANS equations are simplified even further. The exact transport equation for the turbulent kinetic energy, k, can be derived by applying Reynolds decomposition to the instantaneous kinetic energy equation. In the resulting expression, the Reynolds stress terms appear in the production term. These stresses are commonly modeled using the Boussinesq approximation, which relates them to the gradients of the mean velocity field. Using this approach, the production of turbulent kinetic energy can be expressed as:

$$-\langle u_i u_j \rangle \frac{\partial \langle U_i \rangle}{\partial x_j} = \nu_T \left(\frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i} \right) \frac{\partial \langle U_i \rangle}{\partial x_j} - \frac{2}{3} k \frac{\partial \langle U_i \rangle}{\partial x_i}. \tag{2.39}$$

The second closure needed to model the k equation is a relation for the energy dissipation rate which is described by

$$\begin{aligned} \frac{\partial \varepsilon}{\partial t} + \langle U_j \rangle \frac{\partial \varepsilon}{\partial x_j} = C_{\varepsilon 1} \nu_T \frac{\varepsilon}{k} \left[\left(\frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i} \right) \frac{\partial \langle U_i \rangle}{\partial x_j} \right] \\ - C_{\varepsilon 2} \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right]. \end{aligned} \tag{2.40}$$

The turbulent viscosity can now be calculated

$$\nu_T = C_\mu \frac{k^2}{\varepsilon}. \tag{2.41}$$

The five closure coefficients $C_\mu, C_{\varepsilon 1}, C_{\varepsilon 2}, \sigma_k$ and σ_ε are assumed to be universal constants and the values are found in Table 2.1 .

Table 2.1: Closure coefficients in the standard k–ε model.

Constant	Value
C_μ	0.09
$C_{\varepsilon 1}$	1.44
$C_{\varepsilon 2}$	1.92
σ_k	1.00
σ_ε	1.30

The k-ε model assumes that the flow follows the logarithmic law of the wall [32]. Consequently, there are two key areas where the model does not apply. The first

is wall-bounded shear flows that lack a well-defined boundary layer, such as at stagnation points or flow separation regions. The other area is the viscous sub-layer below the log-law layer. The standard k-ε model accounts for this region by applying appropriate boundary conditions for k and ε. However, this approach requires that the first mesh point is positioned within the logarithmic region of the boundary layer, specifically within the range $0 \leq y^+ \leq 500$.

2.4.4.3 Reynolds Stress Model

Turbulence models based on the Boussinesq approximation are inadequate for flows with rapid changes in the mean strain rate, as they fail to capture the history effects of the Reynolds stresses [26]. These effects arise due to the relatively slow momentum exchange between eddies, whereas the Boussinesq approximation assumes that eddies behave like molecules and exchange momentum rapidly. RSM overcome this limitation by abandoning the assumption of isotropic eddy viscosity. Instead, they solve transport equations for each component of the Reynolds stress tensor, $\tau_{ij} = -\rho \langle u_i u_j \rangle$, along with an equation for the dissipation rate of turbulent kinetic energy, ε.

The general form of the transport equation for the Reynolds stresses is

$$\frac{\partial \langle u_i u_j \rangle}{\partial t} + \langle U_k \rangle \frac{\partial \langle u_i u_j \rangle}{\partial x_k} = P_{ij} - \varepsilon_{ij} + \phi_{ij} + d_{ij}, \tag{2.42}$$

where the left-hand side describes the local and convective rate of change of the Reynolds stresses, and the right-hand side contains terms representing production (P_{ij}), dissipation (ε_{ij}), pressure-strain redistribution (ϕ_{ij}) and diffusion (d_{ij}).

To close this equation, models are required for the mean velocity field $\langle U_i \rangle$, the Reynolds stresses $\langle u_i u_j \rangle$, the pressure field $\langle P \rangle$ and the dissipation rate ε. RSM are often referred to as second-moment closures, since only third-order terms and higher require additional modeling. The main advantage of RSM is their ability to naturally capture anisotropy and non-local effects. However, they are computationally expensive, as seven additional partial differential equations must be solved, and the strong coupling between these equations can lead to numerical instability.

2.4.4.4 Large Eddy Simulation

When simulating flows with high Reynolds number, one of the main challenges is the presence of very small length scales and timescales [26]. One way to address this is to filter the equations and only resolve intermediate to large turbulent scales while leaving the smallest scales unresolved. This approach is known as Large Eddy Simulation (LES), where the dynamics of the large scales are computed explicitly. The effects of the non-resolved scale are accounted for with subgrid stress models. The governing equations for LES are obtained by applying a spatial filter to remove the small-scale fluctuations. A generalized filter, with ξ is the spatial integration variable, can be defined by

$$\overline{U_i}(x,t) = \int\int\int G(x-\xi;\Delta)U_i(\xi,t)d^3\xi. \tag{2.43}$$

The filter function, G, keeps values of U_i occurring on scales larger than the filter width Δ. Filtering the Navier–Stokes equations enable a separation between the large turbulent scales $\overline{U_i}$, which are resolved, and the smaller scales u_i, which are modeled. The velocity field has the decomposition,

$$U_i(x,t) = \overline{U_i}(x,t) + u_i(x,t). \tag{2.44}$$

The filtered continuity equation becomes

$$\frac{\partial \overline{U_i}}{\partial x_j} = 0 \tag{2.45}$$

and the filtered momentum equation becomes

$$\frac{\partial \overline{U}_i}{\partial t} + \frac{\partial \overline{U}_i \overline{U}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{P}}{\partial x_j} + \nu\frac{\partial^2 \overline{U}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}, \tag{2.46}$$

with τ_{ij} being the subgrid stress tensor. The subgrid stress tensor is causing closure problems in LES and is therefore often modeled using the Smagorinsky-Lilly model

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2\mu_t \overline{S}_{ij}, \tag{2.47}$$

where $\overline{S}_{ij}$ is the time-averaged strain rate tensor. The subgrid viscosity can be calculated from

$$\mu_t = \rho L_S^2 \left|\overline{S}\right|, \tag{2.48}$$

where

$$|\overline{S}| = \sqrt{2\,\overline{S}_{ij}\,\overline{S}_{ij}} \quad \text{and} \quad L_S = \min(\kappa d,\, C_S V^{1/3}). \tag{2.49}$$

κ is the von Kármán constant, d is the distance to the nearest wall, V is the volume of the computational cell, and C_S is the Smagorinsky coefficient.

2.4.4.5 Implicit Large Eddy Simulation

In LES, the smaller turbulent scales are unresolved and modeled [33]. When using Implicit Large Eddy Simulation (ILES) the influence of the unresolved scales are implicitly incorporated, not requiring the need for explicit modeling.

2.4.5 Passive Scalar

The transport of a species within a fluid can be modeled as a passive scalar [2]. A scalar quantity is advected and diffused by the flow but does not influence the flow field itself. The transport of the passive scalar is governed by convection and diffusion according to the following transport equation:

$$\frac{\partial \phi}{\partial t} + \frac{\partial (\phi v_i)}{\partial x_i} = \frac{\partial}{\partial x_i} \left[\left(\frac{\nu}{Sc} + \frac{\nu_T}{Sc_t} \right) \frac{\partial \phi}{\partial x_i} \right], \tag{2.50}$$

where ϕ is the passive scalar, ν is the kinematic viscosity, and ν_T is the turbulent viscosity. The molecular diffusivity is characterized by the Schmidt number, Sc, while the turbulent diffusivity is described by the turbulent Schmidt number, Sc_t. In stirred tank configurations, typical values for Sc_t range from 0.2 to 0.9.

2.4.6 Evaluating Mixing Time

A transient blending simulation is the most effective way to estimate how long it takes for a tracer to become fully mixed in a fluid [22]. If the tracer has the same properties as the fluid, the flow field does not need to be recalculated. Instead, the tracer can be added after the flow has reached steady state. Since the tracer does not affect the flow, the equations for momentum, continuity and turbulence can be disabled during the mixing simulation. This makes the calculation faster and more stable, as only the transport of the tracer needs to be solved. The mixing time can then be determined by monitoring the average concentration and standard deviation in the tank.

2.4.7 Modeling of Multiphase Flows

Multiphase flows are flows that involve the interaction between two or more distinct phases such as solids, liquids and gases, and are common in both natural phenomena and industrial applications [26]. This section provides an overview of those modeling approaches for multiphase flows in CFD that is relevant for this thesis work.

2.4.7.1 Euler–Lagrange Model

In Euler–Lagrange modeling, the fluid phase is modeled as a continuum by solving the Navier–Stokes equations, while for the dispersed phase a large number of individual particles is modeled. The dispersed phase can exchange momentum, mass, and energy with the fluid phase.

In point-particle-approach simulations, the single-phase Navier–Stokes equations for the continuous phase are solved in conjunction with tracking the individual particles,

$$\frac{\partial (\alpha_f \rho_f)}{\partial t} + \frac{\partial (\alpha_f \rho_f U_{i,f})}{\partial x_i} = S_c, \tag{2.51}$$

$$\frac{\partial(\alpha_f \rho_f U_{i,f})}{\partial t} + U_{i,f}\frac{\partial(\alpha_f \rho_f U_{j,f})}{\partial x_j} = -\alpha_f \frac{\partial P}{\partial x_i} + \frac{\partial(\alpha_f \tau_{ij,f})}{\partial x_j} + S_{i,p}, \tag{2.52}$$

where α_f is the volume fraction of the fluid, ρ_f is the fluid density, $U_{i,f}$ is the i-th component of the fluid velocity and $\tau_{ij,f}$ is the viscous stress tensor. The source term S_c accounts for mass transfer between the phases, while $S_{i,p}$ represents the momentum exchange between the fluid and the particles.

This formulation assumes that the particles are significantly smaller than the grid cells, so that the local fluid velocity used in the force calculation remains undisturbed by the presence of the particles. Euler–Lagrange models are particularly effective at low particle volume fractions with one- or two-way coupling. At higher concentrations, particle–particle interactions become important, requiring additional closures and significantly increasing the computational cost.

2.4.7.2 Volume of Fluid

The volume of Fluid (VOF) model is an Euler-Euler approach that tracks the interface between different phases [26]. VOF works by predicting the position of the interface using a color function γ, which represents the volume fraction of the tracked phase in each cell. If a cell contains none of the tracked phase, $\gamma = 0$. If it is completely filled with the tracked phase, $\gamma = 1$ and if it contains both phases, $0 < \gamma < 1$. While all phases share a common set of momentum equations (one for each spatial direction), the volume fraction is used to describe how much of each phase is present in each cell [26].

In cells with only one phase present, the properties are those of that phase [26]. For cells with more than one phase present, the properties are volume-weighted averages. The volume fraction is modeled using the continuity equation, and the surface tension is added as a source term in the momentum equation in cells where two phases or more are present.

2.4.8 Grid Convergence Index

The Grid Convergence Index (GCI) method is a procedure for estimating the discretization error in a computational grid [34]. The method is based on the Richardson extrapolation method and is a recommended and widely evaluated method within CFD. The Richardson extrapolation is a technique used to improve the accuracy of numerical results by combining solutions at different resolutions [35]. By assuming a Taylor series expansion with respect to the discretization parameter, Richardson extrapolation can eliminate the dominant sources of numerical error, resulting in faster convergence and improved accuracy. A known limitation of Richardson extrapolation is that the predicted values may not vary smoothly or monotonically with grid refinement [34]. In time-dependent simulations, this irregular behavior can also vary across both time and space.

In the GCI method, when considering a 3D case, a representative cell size h is

defined,

$$h = \left[\frac{1}{N} \sum_{i=1}^{N} (\Delta V_i) \right]^{1/3} \tag{2.53}$$

where ΔV_i is the volume of the i:th cell and N is the total number in the computational grid [34]. Three or more different grids are required for the calculations and the refinement factor between the grids are calculated as

$$r_{ij} = \frac{h_i}{h_j} \tag{2.54}$$

where grid j is finer than grid i. The grid refinement should be done systematically even if the grid is unstructured. After running the simulations for each grid, values of a key variable, ϕ, is evaluated for each grid. It is appropriate to use the average global cell size in equation 2.53 if a global variable, such as velocity, is used. Let h_k<hj<h_i. The apparent order p of the method can be calculated as

$$p = \frac{1}{\ln(r_{jk})} \left| \ln \left| \frac{\varepsilon_{ij}}{\varepsilon_{jk}} \right| + q(p) \right| \tag{2.55}$$

$$q(p) = \ln \left(\frac{r_{ji}^p - s}{r_{ij}^p - s} \right) \tag{2.56}$$

$$s = 1 \cdot \operatorname{sgn} \left(\frac{\varepsilon_{ij}}{\varepsilon_{jk}} \right) \tag{2.57}$$

where ε_{ij}=ϕ_i - ϕ_j and ε_{jk}=ϕ_j - ϕ_k. Finally the approximate relative error, e_a^{ij} and the fine-grid convergence index, GCI_{fine}^{ij} can be calculated as

$$e_a^{ij} = \left| \frac{\phi_j - \phi_i}{\phi_j} \right| \tag{2.58}$$

$$GCI_{\text{fine}}^{ij} = \frac{1.25\, e_a^{ij}}{r_{ij}^p - 1}. \tag{2.59}$$

2.5 Software and Hardware

To evaluate and compare the performance of the numerical methods considered in this thesis project, two different computational software have been employed. Each software is based on either FVM or LBM, and therefore utilizes different approaches

when it comes to, for example, turbulence modeling and rotational motion. The following section presents the software tools and required hardware used in this thesis, along with a brief description of their numerical implementation.

2.5.1 M-Star

The M-Star solver is a LBM CFD tool [36]. M-Star has two different lattice types that primarily differ in the number of discrete velocity directions [5]. The two available lattice types are D3Q19 and D3Q27 which allows 19 and 27 possible velocity directions respectively. The additional directions in the latter type allows it to capture more complex flow details. However, this lattice type is more computationally demanding requiring about 40% more memory and runs approximately 40% slower. In a trade off between computational efficiency and accuracy, the D3Q27 type may be preferable in simulations with more complex flow details, higher Reynolds number and complex boundary interactions. The LES turbulence model allows the user to select lattice type, whereas when using the ILES turbulence model, the D3Q27 lattice type is required.

The most computationally efficient solution is the default cuboid lattice with uniform spacing [5]. However, when modeling systems with high aspect ratio features, such as small impellers inside large tanks, local refinement zones may be necessary. Although adding refinement reduces computational efficiency, it can increase the memory efficiency of the code. Local refinement zones are user-defined regions, which can have various shapes, where the lattice undergoes a step change in both spatial resolution and time step size. Multiple refinement levels can be applied, where each level increases the local resolution by dividing the baseline lattice spacing by a factor of two. If the refinement level exceeds two, a set of transition layers is automatically added outside the refined region to gradually reduce the resolution back to the baseline. When using refinement in the simulation setup, the ILES turbulence model is required.

2.5.2 STAR-CCM+

STAR-CCM+ is a FVM-based CFD solver developed by Siemens [37]. It supports a wide range of flow simulations, including incompressible and compressible flows, multiphase systems, heat transfer, and rotating machinery. The software includes several turbulence models, such as RANS, LES, and DES, allowing for flexibility depending on the accuracy and computational cost required.

STAR-CCM+ features automated mesh generation and supports polyhedral, hexahedral, and trimmed cell meshes. Mesh refinement can be applied locally to resolve complex geometries or flow features. The software also includes built-in tools for simulating moving or rotating parts using either MRF or SM approach. For time-dependent problems, both steady-state and transient solvers are available.

2.5.3 CPU and GPU

A Central Processing Unit (CPU) is a microprocessor that executes instructions from a program using arithmetic, control, logic, and input/output operations [38]. Its main components include an arithmetic logic unit, processor registers for storing and handling data, and a control unit that manages the execution of instructions from memory. To reduce response time, CPUs are designed with low latency in mind and include large caches that temporarily store data to avoid slower access to global memory. Most modern computers use multi-core processors, meaning a single chip contains two or more processing units, called cores. These cores allow multiple tasks or processes to be executed in parallel, improving performance in multitasking and parallel computing.

The design and purpose of a Graphics Processing Unit (GPU) differs significantly from that of a CPU [38]. While CPUs are optimized for low latency and efficient processing of sequential instructions, GPUs were originally developed for rendering graphics, where many pixel values can be calculated independently. Today, GPUs are widely used beyond graphics, including general-purpose computing and scientific simulations. Unlike CPUs, GPUs are designed for high throughput, meaning they can process large amounts of data simultaneously. They dedicate more hardware to computation and less to fast cache memory, which is the opposite of a CPU.

A main difference is that a CPU has a few powerful cores, while a GPU consists of thousands of smaller, simpler cores [38]. This makes GPUs well-suited for parallel tasks with low memory demands, while CPUs remain better at handling tasks that are sequential and memory-intensive.

Which hardware platform performs best depends largely on the amount of data being processed and the degree of parallelism available [38]. For smaller workloads, or when only a limited number of parallel instances can be utilized, CPUs tend to be more efficient due to their low overhead and fast memory access. When a high degree of parallelism is available, however, GPUs generally offer better performance. As the computational domain grows, such as in simulations of large geometries, the amount of data increases significantly, which can reduce the efficiency of GPU-based computations due to memory limitations.

3

Methods

In this section, the methodology for the simulation cases is presented. A total of three different cases have been simulated. Each case begins with a short case description, then follows the method for the FVM simulations, and after that the method for the LBM simulations. This structure is repeated in all three cases.

For all LBM simulations, the immersed boundary method is used as default for handling moving parts, such as the impeller [5]. All FVM simulations use the MRF or SM method for all moving parts, how this was implemented for each case will be presented in a more detailed way for each case.

Most of the FVM simulations were run using 64 CPU cores of the type *AMD EPYC 7532 32*. The LBM simulations were instead run on GPU of the type *NVIDIA RTX^{TM} 6000 Ada Generation* with 48GB memory.

3.1 Case 1

The first simulation case involves a biomedical mixing tank with a diameter of 680 mm and an impeller with a diameter of 148 mm. The tank and impeller geometry are illustrated in Figure 3.1.

(a) Geometry of the tank in case 1 in the XY-plane. The tank is cylindrical and the impeller is positioned near the bottom and offset from the center.

(b) Geometry of the Metenova ZG5 magnetic mixer.

Figure 3.1: Geometry of the tank and impeller used in Case 1.

The tank geometry and impeller configuration used in this study are identical to those investigated in the master's thesis by Larsson and Lindenbaum [2], where both experimental and numerical analyzes were performed. Their experimental data form the basis for the validation of the simulations presented in this work. In their study, impeller torque was measured to calculate the power number and mixing time experiments were carried out using saline as a tracer. The mixing time was determined from conductivity measurements taken at multiple locations within the tank, allowing estimation of the time required to reach 95% homogeneity.

To evaluate the numerical methods, two different cases were simulated using this tank geometry. Both cases have also been studied experimentally. The corresponding parameters are presented in Table 3.1.

Table 3.1: Experimental cases with corresponding parameters.

Case	**A**	**B**
Water level [mm]	816	816
Impeller speed [RPM]	268	337
Reynolds number [-]	97 600	123 000

Due to the high Reynolds numbers in both cases, the flow is considered fully turbulent, and a vortex is expected to form at both impeller speeds. Prior to simulation preprocessing, the impeller geometry was slightly simplified using computer-aided design (CAD) by removing sharp edges that would otherwise be difficult to mesh.

3.1.1 FVM simulations - Case 1

For the FVM simulations of case 1, both steady-state and transient approaches were employed. A three-dimensional, segregated solver was used to compute the flow field, with the SIMPLE algorithm applied for pressure–velocity coupling. The fluid densities were assumed constant throughout the simulations. Convergence for the steady-state simulations was considered achieved when the scaled residuals fell below 10^{-4}. In the transient simulations, 15 inner iterations were performed per time step. All tank-walls were considered as no-slip boundaries except for the top surface where a slip-wall boundary condition was applied.

3.1.1.1 Turbulence Model

The selection of a turbulence model in FVM simulations requires careful consideration, as it can significantly impact key parameters such as velocity profiles, impeller power number, and mixing time. To assess the impact of different turbulence models on the results, it is common practice to perform a sensitivity study, comparing multiple turbulence models to determine the most suitable one for accurately capturing the flow behavior.

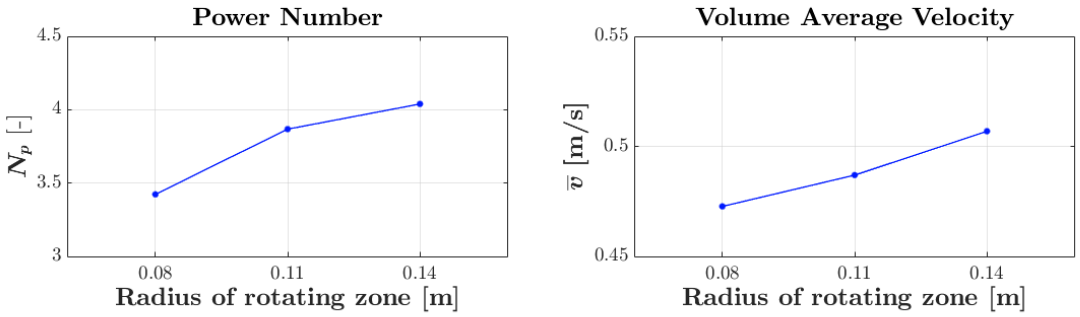
However, based on the work by Larsson and Lindenbaum [2], who studied an identical setup in terms of geometry and impeller speeds, it was shown that among the RANS models, only RSM was capable of accurately capturing the vortex development. Given that RANS turbulence models are generally more computationally efficient than higher-fidelity models such as LES or DES, and that RSM provided sufficient accuracy, it was chosen for the simulations of case 1. Since the focus of this master's thesis is not on turbulence model evaluation, further investigation into turbulence modeling was not included in the study for case 1.

3.1.1.2 Rotational Motion

For the FVM simulations of case 1, the rotational motion of the impeller was modeled using different approaches depending on the simulation type. For the steady-state simulations, MRF was employed. For the transient simulations, SM was used, which explicitly resolves the motion of the rotating domain over time. The time step for the transient simulations was determined based on the constraint that the rotating zone should not move more than 1° per time step, ensuring an accurate representation of the impeller motion while maintaining computational efficiency. This resulted in a time step size of 0.5 ms for the impeller speed of 337 rpm and 0.6 ms for the impeller speed of 268 rpm.

Additionally, a sensitivity study was conducted to evaluate how the size of the rotating zone influenced the simulation results. This analysis was crucial to ensure that the chosen rotational domain was sufficiently large to capture the relevant flow dynamics without introducing artificial effects at the interface between the rotating and stationary regions. In order to determine the most optimal size of the rotating zone, the simulated impeller parameters were compared to experimental data.

Figure 3.2: Positioning of the rotating zone within the tank geometry. This region encloses the impeller and is used in both MRF and SM simulations to model impeller motion.



(a) Power number as a function of rotating zone radius.

(b) Volume-averaged velocity as a function of rotating zone radius.

Figure 3.3: Effect of rotating zone radius on power number and volume-averaged velocity.

The power number, shown in Figure 3.3a, was determined using Equation 2.4. Since the impeller speed and diameter are known, the only required variable for the calculation was the impeller torque. The volume average velocity, shown in Figure 3.3b, is the volume average velocity throughout the whole tank. Thus, it was concluded that a rotating zone with a radius of 0.11 meters were most similar to the experimental data of the power number obtained in the master's thesis by Larsson and Lindenbaum [2]. The experimental data is presented in chapter 4.

3.1.1.3 Mesh

The volume mesh was generated using a polyhedral mesher with additional prism layers and surface remesher. For wall functions the Two-Layer All y+ Wall Treatment was applied for all meshes. Prism layers were applied to the tank walls and the impeller. The number of prism layers for the tank walls was set to 6 and 7 for the impeller. The size of the prism layers was chosen so that the y+ was below 5 for the tank walls and below 1 for the impeller. Furthermore, two additional prism layers were applied, one on each side of the interface, based on recommendations from STAR-CCM+ for meshing a cylindrical interface between a rotating and a stationary region [39].

All meshes tested for case 1 had the interface target surface size set to 50% of the maximum cell size and the cell size in the MRF region set to 25% of the maximum cell size. The growth rate was set to 1.2 to ensure a slow transition from the finer mesh areas.

3.1.1.4 Mesh Independence Study

To ensure the accuracy and reliability of the CFD simulations, a mesh independence study was conducted using the RSM turbulence model with a single-phase setup. Only steady-state solutions were considered throughout the study, and the study was performed at only one impeller speed, 337 RPM. The mesh study focused on evaluating the effect of mesh resolution on a few key performance parameters, which were the impeller power number, pumping number, volume-averaged velocity and the velocity magnitude along a probe placed in the middle of the tank. To assess the rate of convergence, the GCI method was applied with a tolerance of 1% for the key parameters.

The mesh independence study was based on previous work by Larsson and Lindenbaum [2], which indicated that at least 1.5 million cells were necessary to obtain reliable results. Therefore, the study was initiated at this resolution, and the number of cells was progressively increased. The cell count was increased by a factor of approximately 1.2–1.3 between each mesh. Refinement was performed by adjusting the maximum cell size im the volume mesh and the target size on the impeller surface, which increased the total number of cells while maintaining a structured refinement approach. The number and thickness of the prism layers were kept constant throughout the mesh independence study. Table 3.2 presents the different mesh configurations that were tested.

Figure 3.6 shows the variation of the chosen parameters with an increasing number of cells. The pumping number, shown in Figure 3.4b, was calculated using Equation 2.6, where N is the impeller speed in revolutions per second and D is the impeller diameter. Thus, the volumetric flow rate was the only remaining parameter to be obtained from the simulations. To accurately measure the volumetric flow rate around the impeller, a cylindrical surface with a diameter of 150 mm (2 mm larger than the impeller) was placed around it. Larsson and Lindenbaum examined the effect of cylinder height and concluded that a height of 50 mm is sufficient to

Table 3.2: Mesh settings used in the mesh independence study for Case 1, including total cell count, maximum cell size, and target size on the impeller surface.

Number of cells [million]	Maximum cell size [mm]	Impeller surface target size [mm]
1.53	20	0.9
2.04	15	0.8
2.68	10	0.8
3.49	8.0	0.7

capture the volumetric flow rate [2].

Furthermore, the velocity magnitude was measured along a probe placed in the middle of the tank for each tested mesh. The result is shown in Figure 3.5.

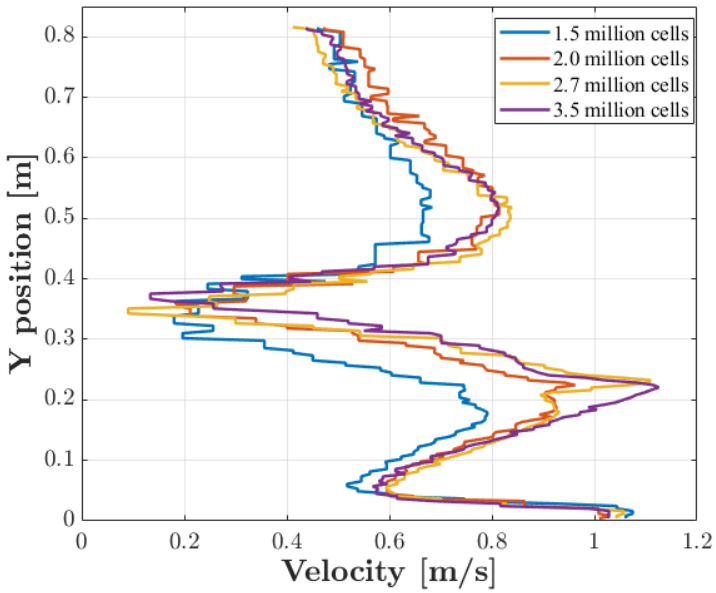
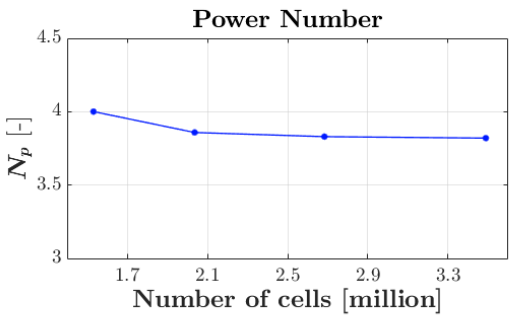


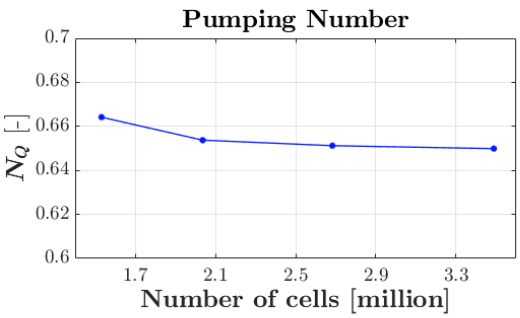
Figure 3.5: Velocity magnitude along a central vertical probe in the tank for different mesh resolutions. The probe is positioned at the center of the tank and aligned with the vertical axis.

The GCI obtained from the mesh with 2.0 million cells was below the set 1% tolerance for all evaluated parameters. Together with the convergence of the power number and pumping number shown in Figure 3.4, this indicates that a mesh-independent solution is achieved with approximately 2.0 million cells. Furthermore, the velocity magnitude measured along the probe, shown in Figure 3.5, reinforces this conclusion.

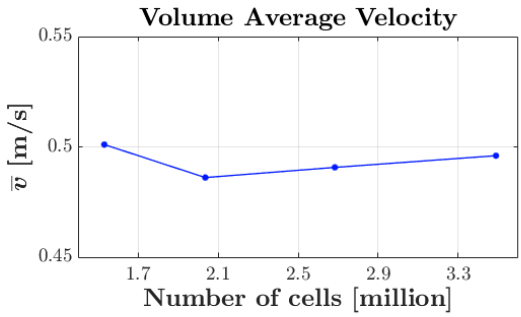
Figure 3.6 presents the final mesh with about 2 million cells, were Figure 3.6a shows the meshed tank in the XY-plane while Figure 3.6b shows a close-up of the mesh of



(a) Power number as a function of number of cells.



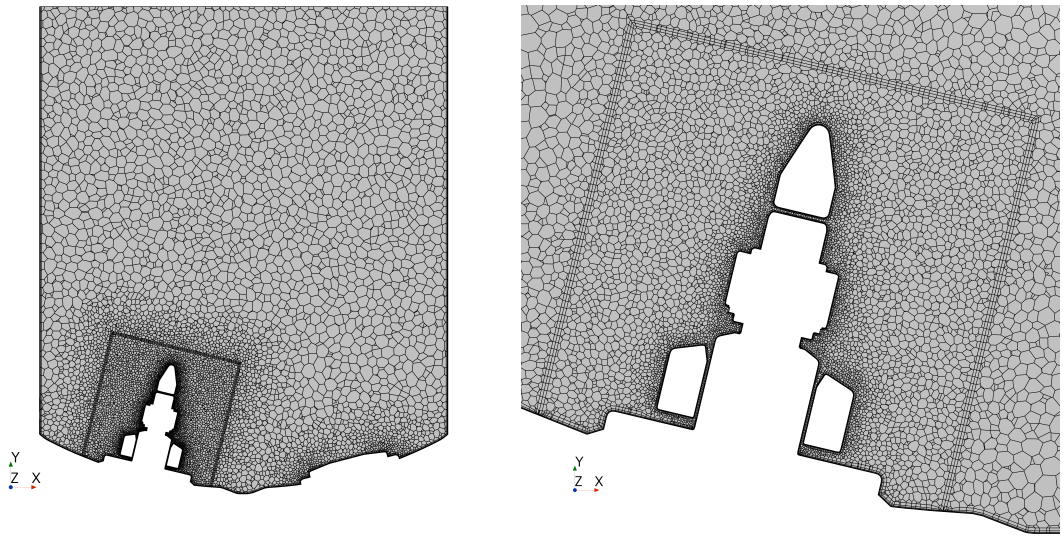
(b) Pumping number as a function of number of cells.



(c) Volume average velocity as a function of number of cells.

Figure 3.4: Mesh independence study for the FVM simulations in Case 1 with power number, pumping number, and volume average velocity analyzed as functions of the number of cells.

the MRF zone in the XY-plane.



(a) Cross sectional mesh configuration in the XY-plane over the tank.

(b) Cross sectional mesh configuration in the XY-plane around the impeller.

Figure 3.6: Mesh configuration for final mesh in Case 1.

3.1.1.5 Mesh Refinement for VOF

To accurately capture the formation of the vortex, the mesh was refined again once the VOF model was activated. The mesh was applied to the top of the tank as well, where the air was located. The cell size was reduced to 4 mm at the water surface and 2 mm in the area where the vortex was predicted to form. At an impeller speed of 268 RPM, the refined mesh resulted in approximately 4.2 million cells. By observing the formation and placement of the vortex in the simulation with the lower impeller speed, a more precise refinement area could be identified. By reducing the refinement area, the cell count could be reduced to approximately 3.5 million. To save computational time, the mesh with a smaller refinement area was used during the simulations with higher impeller speed.

Since this mesh size was shown to be accurate enough to capture the vortex in the earlier work by Larsson and Lindenbaum, which used the same geometric setup, no further mesh refinement study was performed [2]. Figure 3.7 shows the refined mesh in the XY-plane.

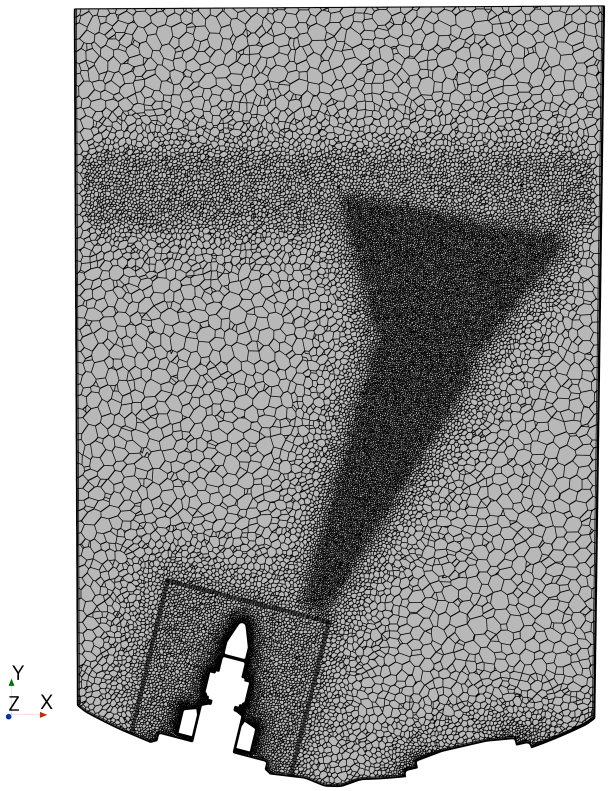


Figure 3.7: Cross sectional mesh configuration in the XY-plane with refinement for VOF.

3.1.1.6 Capturing the Vortex

To reduce the computational time, the VOF simulations were initialized using a previously computed steady-state flow field. A surface tension force of 0.072 N/m was included in the VOF model. Once the vortex had formed and stabilized, an iso surface was created based on the volume fraction of water. This isosurface was then treated as a geometry part and excluded from the fluid domain. By doing this, the vortex could be handled as a slip-wall boundary, which allowed for faster and more stable mixing time evaluations. The same approach was used in previous studies by Larsson and Lindenbaum and Pekkulla et al. [2, 6]. An overview that summarizes the method is illustrated in Figure 3.8.

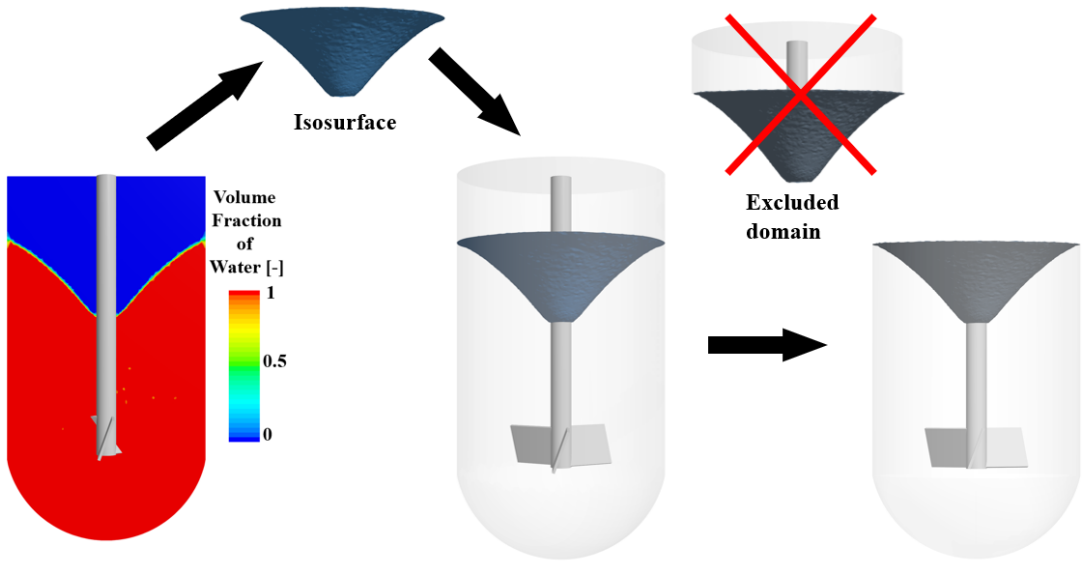
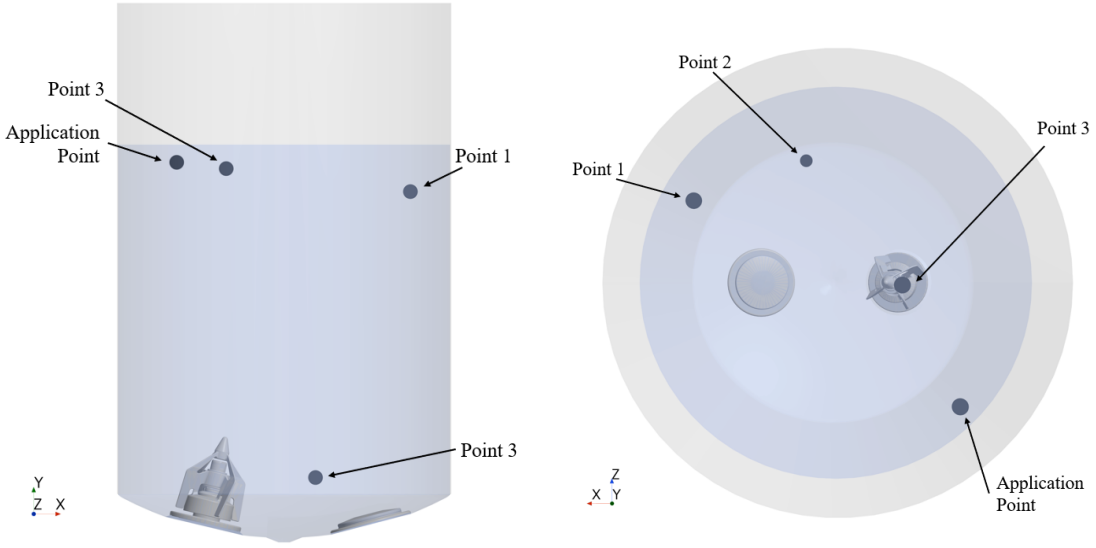


Figure 3.8: Overview of the method used to capture the vortex in FVM, enabling the vortex to be modeled as a slip-wall boundary. An isosurface representing the air–water interface is extracted from the volume fraction field (left), and the domain above this interface is excluded to simplify the geometry. The tank and vortex shown is simulated in Case 3, where the same approach is applied as in Case 1.

3.1.1.7 Determining Mixing Time

To estimate the mixing time a passive scalar was used. 100 cm^3 tracer was injected at the application point, shown in Figure 3.9, on a frozen flow field obtained from steady simulations, to reduce the computational effort. The properties of the tracer were determined by the Schmidt number and the turbulent Schmidt number. The Schmidt number was set to 450 based on self-diffusion of water and the turbulent Schmidt number to 0.7.



(a) Measurement points and application point for the tracer in the XY-plane.

(b) Measurement points and application point for the tracer in the XZ-plane.

Figure 3.9: Location of measurement points and tracer application point used for the mixing time analysis.

The tracer concentration was measured at the three points shown in Figure 3.9. The three points were positioned to match the probe locations used in the experimental setup. Mixing was considered complete when the normalized tracer concentration at all three measurement points remained within the range 0.95 to 1.05. In addition, CV was calculated based on the standard deviation and the mean tracer concentration. Mixing was considered complete when the CV dropped below the threshold value of 0.0283.

3.1.2 LBM simulations - Case 1

Below follows a brief description of the simulation method used for the LBM simulation of case 1. To enable a fair comparison, the setup was designed to resemble the FVM simulation as closely as possible.

3.1.2.1 Models

In order to effectively capture the flow field and the formation of the vortex, the free surface model was applied in M-Star. It was primarily chosen over the immiscible two fluid model because it is more computationally affordable since only the liquid phase is solved, and the gaseous phase (air) is treated as a background fluid. Simulations with the immiscible two fluid model were run in order to ensure that the free surface model did not exclude any accuracy regarding the formation and movement of the vortex. However, since it was found that the two models gave similar results, the immiscible two fluid model was not investigated further due to the increase in simulation time compared to the free surface model.

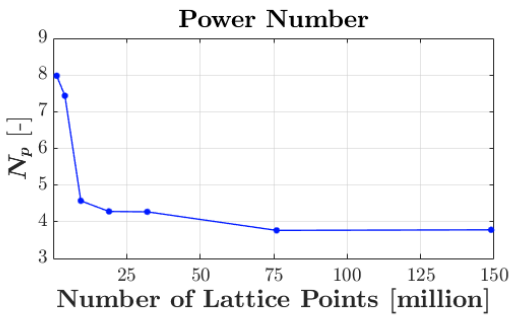
Furthermore, a three-dimensional, transient solver was used to compute the flow field. The LES turbulence model was applied together with VOF in order to model the interface, on a D3Q19 grid. A surface tension force was included with a constant value of 0.072 N/m. Once the vortex had stabilized, a passive scalar was inserted at the application point shown in Figure 3.9. The concentration of the tracer was then measured in the three probes shown in Figure 3.9, and the mixing time was also determined by measuring CV in an equal way as in the FVM simulations.

The time step was determined on the basis of the Courant number, which was set between 0.01 and 0.1 for the simulations, according to recommendations in the M-Star documentation. Furthermore, the dependence of timestep was investigated by running simulations with two different timestep sizes. The chosen time step sizes were based on the Courant number, which were set to 0.05 and 0.01. This corresponds to a timestep size of 66 and 32 μs, respectively. Although the time step size was decreased by 50%, the effects on the impeller parameters were minimal, so it was determined that a time step size of 66 μs could be used to save computational time.

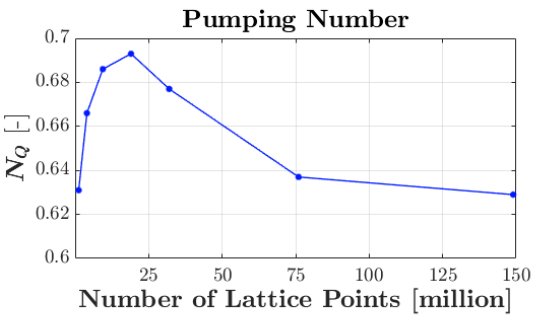
3.1.2.2 Resolution Independence Study

To ensure the accuracy and reliability of the CFD simulations, a resolution refinement study was conducted using the LES turbulence model with free surface modeling of the interface. According to the M-Star documentation, achieving an accurate prediction of shaft power and power dissipation in the fluid requires between 80 and 100 lattice points per impeller diameter [5]. This corresponds to a resolution of approximately 400 to 500 lattice points across the tank diameter. A higher resolution than 500 lattice points across the tank diameter was not possible to achieve since that would require more memory than available on the GPU. However, lower resolutions were also included in the study to provide a complete evaluation of their impact on key parameters.

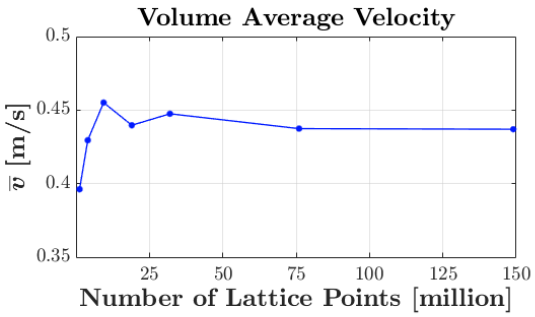
The parameters analyzed in this resolution dependence study were the same as those examined in the mesh independence study for FVM, ensuring a fair comparison. However, since M-STAR is inherently a transient solver, only fully transient solutions were performed in this study. Additionally, the resolution independence study was carried out exclusively for the higher impeller speed of 337 RPM. The power number and pumping number were determined in the same way as in the FVM simulations, described in Section 3.1.1.4.



(a) Power number as a function of number of lattice points.



(b) Pumping number as a function of number of lattice points.



(c) Volume average velocity as a function of number of lattice points.

Figure 3.10: Resolution independence study for the LBM simulations with power number, pumping number, and volume average velocity analyzed as functions of the number of lattice points.

Figure 3.10 shows the result for the chosen parameters measured in the resolution study. It was determined that a grid with about 75 million cells was sufficient, which corresponds to 400 lattice points across the tank diameter.

3.2 Case 2

The second simulation case involves a wastewater digester mixer. This digester features mechanical mixing with side agitation using a draft tube and a freewheeling impeller. The tank has a diameter of 18 m, a height of 9.85 m and a volume of 1932 m^3. The tank geometry and the coordinate system axes are shown in Figure 3.11. The origin is defined at the center of the tank bottom, excluding the lowered section. Both impellers are oriented along the Y-direction and are positioned in a horizontal

plane where z = 0.075 m. The draft tube opening is positioned 25 cm below the top surface. The freewheeling impeller is referred to as impeller 1, while the impeller driving the flow through the draft tube is referred to as impeller 2.

Figure 3.11: Geometry of tank in case 2.

The fluid is a type of sludge, modeled as a non-Newtonian fluid with shear-thinning properties. Considering the complexity of sludge, certain assumptions and simplifications are necessary to model its characteristics. Since sedimentation in digester occurs over a timescale of years, while the simulation time in this case is only one hour, the effect of sedimentation can be neglected. As the trajectories of individual debris are not relevant to the analysis, the sludge is treated as a single phase as often applied according to previous research. The Herschel-Bulkley model is adopted to describe the rheology of the fluid according to Equation 2.2. To prevent non-physical divergence at low shear rates and unrealistically low apparent viscosity values at high shear rates, the model introduces lower and upper boundaries for the apparent viscosity, denoted as μ_{min} and μ_{max}, respectively. Case 2 is further divided into two simulation scenarios, each characterized by fluids with different yield stress thresholds and consistency factors referred to as fluid A and B. The fluid properties for the two different fluids simulated in this case can be found in Table 3.3.

Table 3.3: Fluid properties for fluid A and fluid B

Parameter	Fluid A	Fluid B
$\mu_{\min}$ $[Pa \cdot s]$	0.001	0.001
$\mu_{\max}$ $[Pa \cdot s]$	1000	1000
k [-]	1.134	4
μ_0 $[Pa \cdot s]$	1000	1000
τ_0 $[Pa]$	1.377	1.8
n [-]	0.3	0.3
ρ $[kg/m^3]$	1000	1000

3.2.1 FVM Simulations - Case 2

A three-dimensional, steady, segregated solver was used to compute the flow field. The SIMPLE pressure–velocity coupling algorithm was applied. The sludge properties were modeled using a non-Newtonian generalized power law for the dynamic viscosity, assuming constant density. All tank walls and the bottom were defined as no-slip boundaries, while the lid was set to free-slip to replicate a free surface condition. The rotational speed of both impellers was fixed at 450 rpm. Convergence was assumed when the scaled residuals dropped below $1 \cdot 10^{-4}$.

3.2.1.1 Turbulence Model

The standard k–ε turbulence model was used for both fluid cases, with standard wall function approximations applied near the walls. The choice of turbulence model was based on previous research.

3.2.1.2 Rotational Motion

The rotational motion for the FVM simulations was modeled using the MRF method. Impeller 2 is located inside the draft tube structure, with the blades positioned within a pipe with a diameter of 0.35 m, which was applied as the MRF cylinder diameter combined with a length of 0.2 m. Due to the complex geometry surrounding Impeller 2, the size of its MRF zone is highly constrained. Consequently, the MRF zones for both impellers were kept minimal, focusing primarily on the regions around the blades. Both MRF zones are shown in Figure 3.12. As the impeller shafts are located outside the defined rotating zones, a local rotation rate was assigned to capture their rotational motion accurately. A zone with a constant diameter but twice the length (0.4 m) was tested to ensure that the results were independent of the MRF zone size, which the simulation confirmed.

(a) Rotating zone for impeller 1. (b) Rotating zone for impeller 2.

Figure 3.12: Positioning of the rotation zones for both impellers in the tank geometry.

3.2.1.3 Mesh

The core volume mesh was generated using a polyhedral mesher with additional prism layers and surface remesher to accurately capture the flow characteristics near the walls. Five prism layers were applied to all walls. The size was set to satisfy the $y+$ requirements for the chosen k-ε turbulence model. Mesh refinements were applied in regions with large gradients, primarily within the MRF zones, as the impeller areas are the main sources of turbulence. The custom cell size in these zones was maintained at 8% relative to the base size. To ensure a smooth transition between zones, the custom size at the interface was set to 50% of the base size. Two prism layers were applied, one on each side of the interface. The growth rate was set to 1.2 to ensure a slow transition from the finer mesh areas. Eight prism layers were applied for the impeller surfaces to keep the $y+$ for the impellers below 3 to ensure that the near-wall region was well resolved.

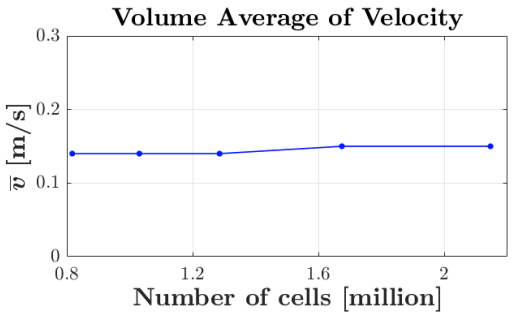
3.2.1.4 Mesh Independence Study

To ensure that the numerical results were not significantly affected by spatial discretization, a systematic mesh convergence study was performed for the setup with fluid A. Five different mesh configurations were evaluated, with a refinement ratio, r, ranging from around 1.2 to 1.3. The mesh configurations used in the study are presented in Table 3.4. To assess the rate of convergence, the GCI method was applied with a tolerance of 1% for the volume-averaged dynamic viscosity and the volume-averaged dynamic viscosity standard deviation, as viscosity in a non-Newtonian fluid is the most challenging parameter to converge.

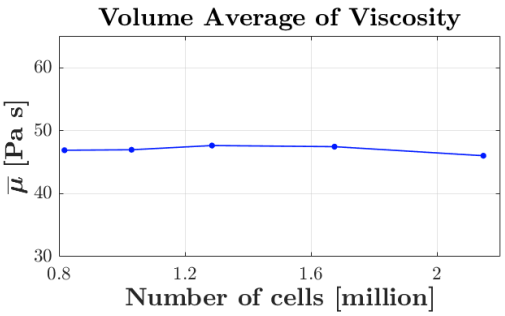
Table 3.4: Mesh parameters used in the core mesh study.

Base size [mm]	Custom size [mm]	Number of cells [million]
177	14.2	0.823
160	12.8	1.03
145	11.6	1.29
130	10.4	1.67
118	9.44	2.15

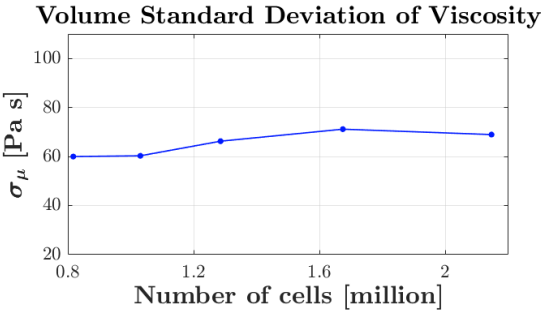
A visualization of the measured parameters as a function of the number of cells is shown in Figure 3.13.



(a) Volume average of velocity as a function of number of cells.



(b) Volume average of viscosity as a function of number of cells.

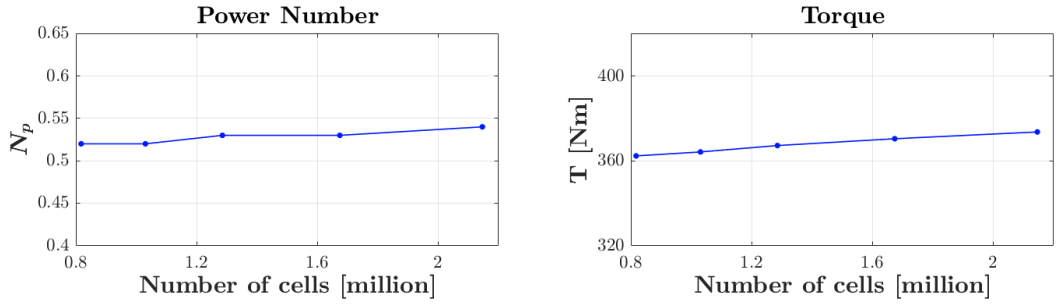


(c) Volume starndard deviation of viscosity as a function of number of cells.

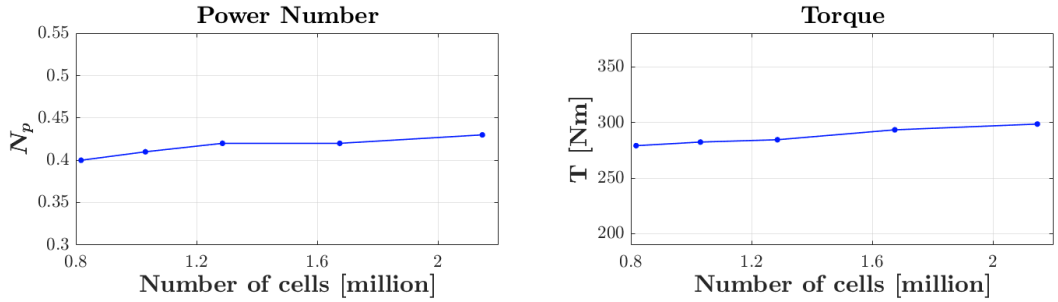
Figure 3.13: Core mesh study for case A with three tank parameters analyzed.

Parameters measured in the MRF zone, such as power number and torque, were

also analyzed during the mesh study. As these parameters converged within the main mesh study, no additional refinement study was required for the MRF region. A visualization of the measured parameters as a function of the number of cells is shown in Figure 3.14.



(a) Power number an torque measured for impeller 1 as a function of number of cells.



(b) Power number an torque measured for impeller 2 as a function of number of cells

Figure 3.14: Core mesh study for case A with two key MRF parameters analyzed.

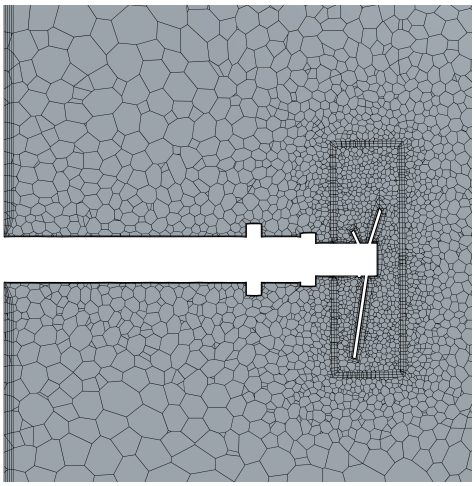
The analysis showed a decreasing relative difference in the measured parameters as the cell size decreased. The mesh with 1.29 million cells exhibited strong agreement with the values obtained for the two finer meshes, and the GCI^{ij}_{fine} value dropped below the tolerance, confirming a mesh-independent solution.

To ensure the numerical accuracy of Case B with fluid properties stated in Table 3.3, the same mesh study used for fluid A was adopted. Since the parameters in the MRF zone also converged within the main mesh study, no additional refinement study was necessary. The GCI values dropped below 1% at the same mesh resolution as in the previous study, confirming mesh independence. The GCI^{ij}_{fine} values for fluid A and fluid B are presented in Table 3.5.

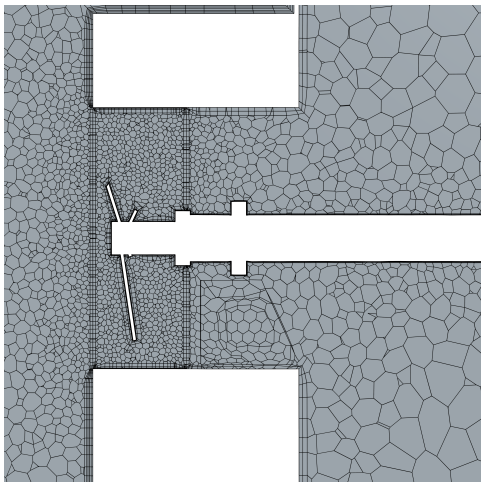
Table 3.5: GCI analysis for Case A and Case B with volume average viscosity and volume average standard deviation as parameters for the selected mesh.

Case A		**Case B**	
Parameter	GCI^{ij}_{fine}	**Parameter**	GCI^{ij}_{fine}
Viscosity	0.800	Viscosity	1.29e-15
Viscosity std	1.75e-5	Viscosity std	9.13e-20

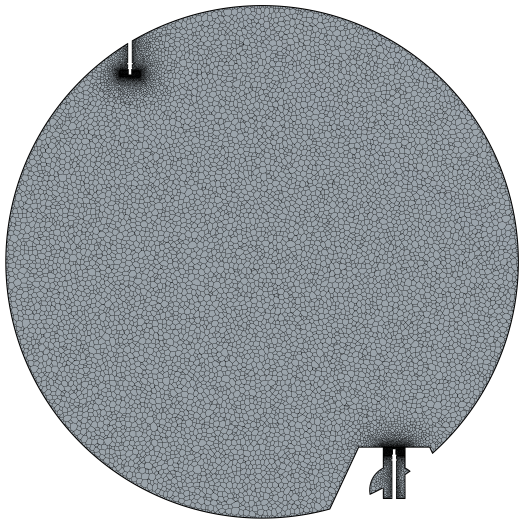
The structure of the selected mesh with refinements in the MRF regions is displayed in Figure 3.23 with a clear boundary for the interfaces between rotating and static region.



(a) Cross sectional mesh configuration in the XY-plane for rotation zone around impeller 1.



(b) Cross sectional mesh configuration in the XY-plane for rotation zone around impeller 2.



(c) Cross sectional mesh configuration in the XY-plane over the tank.

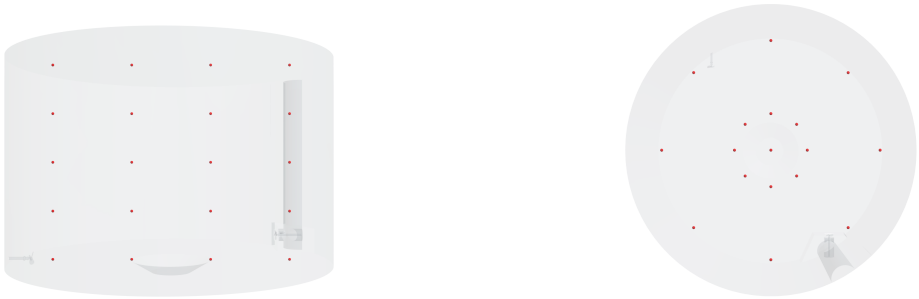
Figure 3.15: Mesh configuration for final mesh in case 2.

3.2.1.5 Determining Mixing Time

To evaluate the mixing time for both fluid cases, a tracer was modeled as a passive scalar and introduced as a $1\,m^3$ cylindrical volume within the draft tube at the top of it. The initial tracer position is shown in Figure 3.16.

Figure 3.16: Initial tracer fluid location on a plane intersecting the center of the draft tube.

To evaluate the mixing time, 85 probe points were distributed throughout the tank to measure the normalized concentration of the tracer. The probe points were arranged across five planes at $z = 0.02$, 2.46, 4.93, 7.39, and 9.85 m. On each plane, 17 points were positioned at $r = 0$, 2, and 6 m, with an angular spacing of 45°, allowing estimation of the time required to reach 95% homogeneity. The positions of the planes across different height levels in the tank, as well as the distribution of probe points over a plane, are shown in Figure 3.17.



(a) Distribution of probe points across four different heights, $z = 0.02$, 2.46, 4.93, 7.39, and 9.85m

(b) Distribution of probe points over each plane at $r = 0$, 2, and 6 m with an angular spacing of 45°

Figure 3.17: Distribution of probe points across the tank for measuring of tracer concentration.

For the mixing time simulations, an implicit unsteady solver algorithm was used instead of a steady-state approach. The tracer fluid was modeled as a passive scalar with convection-only transport. The time step size was set to $5.18 \cdot 10^{-2}$ s in order to maintain a Courant number of 0.05, based on the volume-averaged tank velocity

and the base mesh size for fluid A. To achieve the same Courant number when using fluid 2, the time step size was increased to $8.06 \cdot 10^{-2}$ s due to the lower mean velocity in the tank. For both cases, a simulation with a Courant number of 0.01 was conducted to verify time step independence. As the results showed good agreement with those obtained using a larger time step, a Courant number of 0.05 was selected for the final simulations. The passive scalar was introduced into a frozen flow field obtained from a converged steady-state solution. The maximum number of inner iterations was set to 15. The tracer was considered as mixed when the dimensionless concentration was within a range between 0.95 and 1.05 in all 85 probe points.

3.2.2 LBM Simulations - Case 2

Below follows a brief description of the simulation method used for the LBM simulation of case 2. Similar to case 1, the setup was designed to resemble the FVM simulation as closely as possible.

3.2.2.1 Models

A three-dimensional, transient solver was used to compute the flow field. All tank walls and the bottom were defined as no-slip boundaries, while the lid was set to free-slip to replicate a free surface condition. A scalar field model was used to evaluate the mixing time. The scalar was injected at the same location as in the FVM simulations, and the dimensionless concentration was measured at probe points positioned identically. The scalar injection was initiated after the flow field had reached convergence and the entire simulation was carried out in transient mode.

3.2.2.2 Resolution Independence study

To ensure a lattice independent solution, a uniform resolution dependence study was conducted without any local refinement. The resolution, defined as the amount of lattice points across the tank diameter, was varied. Since the flow around the impellers is critical for mixing in the tank, and these values converged much faster than global parameters such as mean velocity, the study focused on impeller related parameters. The uniform lattice study showed that the highest resolution achievable without exceeding the capacity of the GPU memory was a resolution of 700 lattice points, corresponding to a total of 224.11 million lattice points in the domain. The LES turbulence model and the D3Q19 lattice type was used. The time step size was based on the Courant number, kept at 0.1. The results from the simulations are presented in Table 3.6, where index 1 and 2 represent impeller 1 and impeller 2.

Table 3.6: Uniform lattice study

Resolution	**300**	**500**	**700**
dx [m]	0.053	0.032	0.022
Lattice points [million]	17.6	81.7	224
N_{P1} [-]	4.30	2.31	1.48
N_{P2} [-]	1.71	2.00	1.66
T_1 [Nm]	2920	1590	999
T_2 [Nm]	1180	1370	1130

Since the values did not converge in the uniform resolution study, a refinement study was conducted to accurately resolve the flow around both impellers. The ILES turbulence model for the simulations since this was the required turbulence model in M-Star when using a lattice refinement. The courant number was kept constant at 0.1.

To verify that the results were not sensitive to the size of the refinement zone, a size independence study was carried out using a resolution of 200 lattice points and a refinement level of 2 in the refinement zone. In all simulations, the refinement zone had a cylindrical shape and the Courant number was kept constant at 0.1. Three different lengths were tested while maintaining a fixed diameter of 0.35 m, corresponding to the diameter of the pipe in which impeller 2 is located. The tested lengths were 2, 3 and 4 m, measured from the connection point between the shaft and the wall. The resulting torque and power number values for both impellers showed negligible differences between the 2 m and 3 m cases. Based on these findings, the length was fixed at 2 m for the remaining tests, and the diameter of the refinement zone was varied. The tested diameters were 0.35, 0.5, 0.75, and 1 m. The results showed that the solution became independent of the refinement zone diameter from 0.75 m and above. Based on these results, the final refinement zone was set to a diameter of 0.75 m and a length of 2 m.

To evaluate the convergence of the impeller parameters with applied refinement, a refinement study was conducted where the spatial resolution at the impellers, dx_r, was reduced by a factor of two between each configuration. Due to hardware limitations, it was not possible to achieve a resolution finer than $dx_r = 0.005$ without exceeding the capacity of the GPU memory. The configurations for the tested resolutions are listed in Table 3.7, and the corresponding results are presented in Table 3.8.

Table 3.7: Refinement mesh setups and spatial resolution.

	Resolution	**Refinement level**	$\boldsymbol{dx_r}$ **[m]**	**Lattice points [million]**
r1	200	2	0.020	4.56
r2	200	3	0.010	9.76
r3	200	4	0.005	17.3

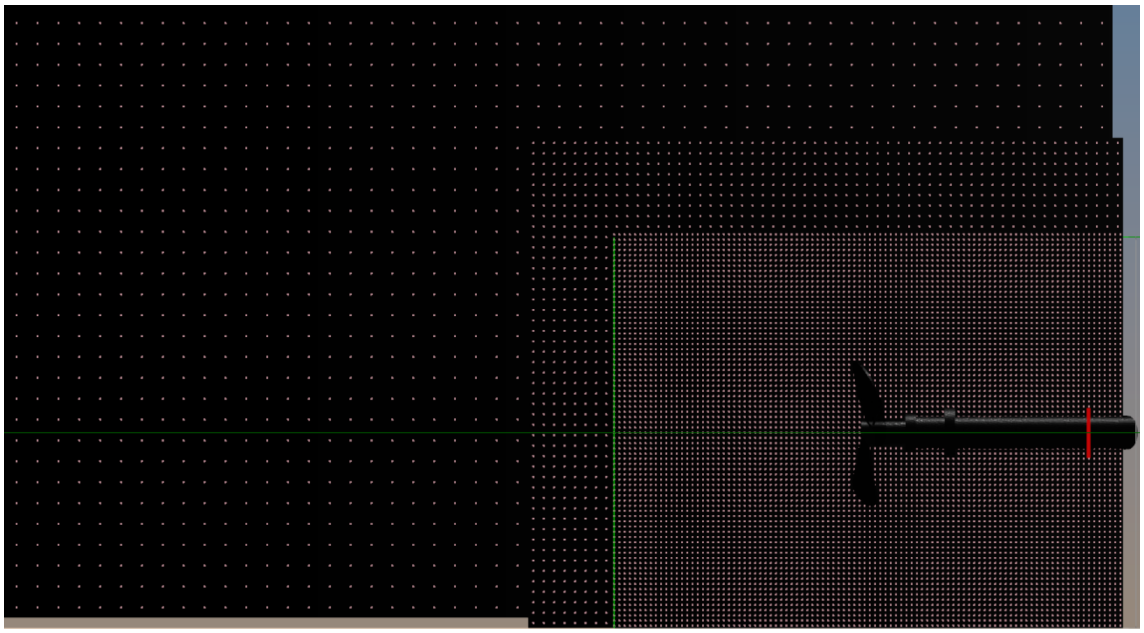


Figure 3.18: Lattice structure with refinement around the impeller and a transition layer from the refinement zone and the main lattice.

Table 3.8: Results from refinement study.

	N_{P1} [-]	N_{P2} [-]	T_1 [Nm]	T_2 [Nm]
r1	1.19	1.23	810	841
r2	0.608	0.767	415	520
r3	0.597	0.605	408	412

Due to the increased size of the refinement zone, the computational cost of the simulations also increased. An estimation of the simulation time required to reach 4000 seconds of physical time was conducted for each configuration, since 300-400 s is required for the flow field to converge and 3600 s for mixing. The estimated run times for configurations r1, r2, and r3 were approximately 30, 130, and 840 hours, respectively. Considering the limited time frame of this thesis, configuration r1 was selected for the mixing time simulations. The structure of the lattice at a plane through impeller 1 where, including the refinement zone and transition layer, is presented in Figure 3.18. Since reducing the Courant number significantly increased the simulation time, a time step sensitivity analysis was not performed for the mixing time study.

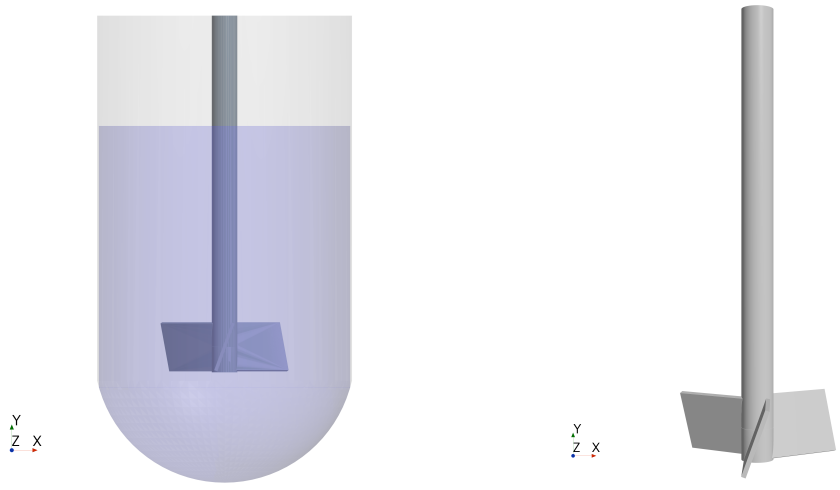
3.2.3 Additional Simulations

Since no other simulation setup was feasible to run for the mixing time simulation, considering the time constraint, some shorter diagnostic simulations were performed to investigate which factors influenced the results. All simulations were run until both velocity and viscosity had converged.

The tested configurations included a setup with configuration r2 (see Table 3.7), a uniform lattice, and a reduced timestep. The uniform lattice setup used an x-resolution of 700 lattice points as described in Table 3.6. Since no refinement was used in this setup, the LES turbulence model was chosen which allowed using the D3Q19 lattice type that increases the computational efficiency. A simulation with a Courant number of 0.05, half the size of the used Courant number, was also conducted using configuration r1.

3.3 Case 3

The tank investigated in case 3 is a type of biomedical tank similar to the tank in case 1. However, compared to the tank in case 1, this setup represents a more typical configuration for mixing particles. The tank in case 3 has a diameter of 0.3 meters and a height of 0.55 meters, with the water level at 0.42 meters. A pitched blade impeller with a 70° blade angle is used, and the impeller diameter corresponds to half the tank diameter. The geometry of the tank and the impeller is presented in Figure 3.19. The rotation of the impeller creates a vortex at the water surface, making this a more complex mixing case. As a result, the system needs to be modeled as a three-phase flow, with gas, liquid, and solid phases simulated simultaneously.



(a) Geometry of Tank 3 in the XY-plane. The tank is cylindrical and equipped with a centrally mounted impeller.

(b) Geometry of the pitched blade impeller used in Tank 3, with a blade angle of 70° and a diameter equal to half the tank diameter.

Figure 3.19: Geometry of the tank and impeller used in Case 3.

Table 3.9: Operational and particle-related parameters for the biomedical tank configuration simulated in Case 3.

Parameter	Value
Water level [mm]	420
Impeller speed [RPM]	173
Particle Diameter [mm]	0.08
Particle Density [kg/m^3]	1970
Volume fraction of particles	0.005

In order to validate the numerical approaches, the results obtained from both simulation softwares will be compared with experimental data from the work by Shan et al., who conducted experiments in an identical tank configuration [1]. The aim is to evaluate how well the particle distribution in the mixing tank is represented. Table 3.9 shows the properties of the tank, particles and fluids.

3.3.1 FVM Simulations - Case 3

For the FVM simulations, a three-dimensional, segregated solver was used to compute the flow field. The SIMPLE model was used for pressure-velocity coupling. The fluid densities were assumed constant throughout the simulations. The simulations were considered to be converged when the residuals for continuity and momentum were below $1 \cdot 10^{-4}$. The final simulations were run with a transient solver, however, the mesh independence study were performed with a steady solver in order to save computational time. For the transient simulations, the inner iterations were limited to 15 per time step. The suitable time step size was determined to be about 0.001 s, according to the rule of thumb for SM that the rotating zone should move about 1 degree per time step.

3.3.1.1 Rotational Motion

Similar to the FVM simulations in case 1 and 2, a sensitivity study was conducted to evaluate how the size of the rotating zone influenced the simulation results. In order to ensure that the results were independent of the size of the rotating zone, a few relevant parameters were measured for different radii of the zone. The location and shape of the cylindrical rotating zone used in these simulations is shown in Figure 3.20.

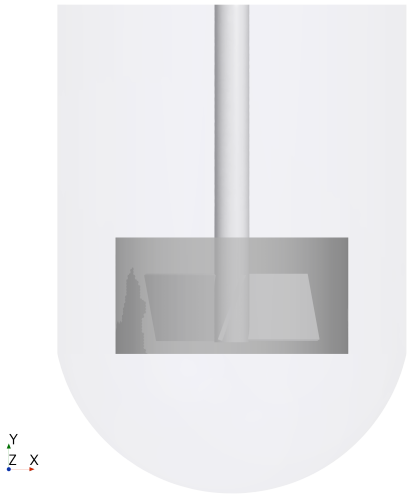
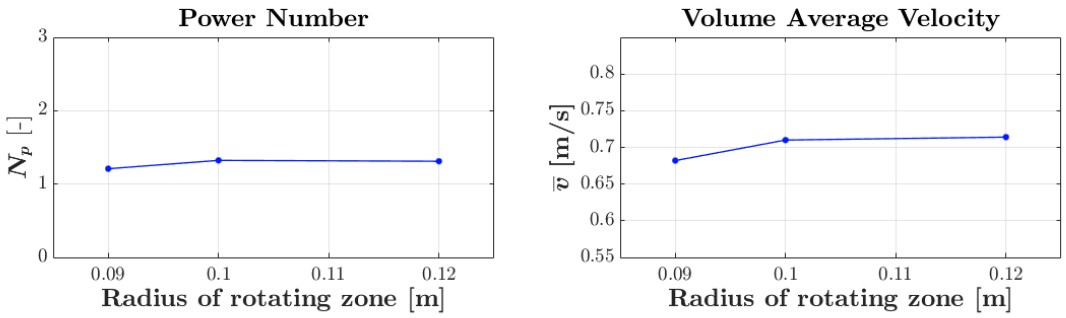


Figure 3.20: Positioning of the rotating zone within the tank geometry for Case 3.

As seen in Figure 3.21, increasing the radius of the rotating zone from 0.10 m to 0.12 m had a negligible impact on both the power number and the volume-averaged velocity. Based on these findings, a radius of 0.10 m was used in the final simulations. Additionally, test simulations were conducted where the height of the cylindrical rotating zone was increased from 0.10 m to 0.12 m. These tests also confirmed that the height variation had no noticeable effect on the results.



(a) Power number as a function of rotating zone radius.

(b) Volume-averaged velocity as a function of rotating zone radius.

Figure 3.21: Effect of rotating zone radius on power number and volume-averaged velocity in Case 3.

3.3.1.2 Turbulence Model

Given the relatively small particle size, the particles are expected to be influenced by the turbulent structures in the flow. Previous studies have shown that the k–ε turbulence model provides results of sufficient accuracy for similar mixing applications [1, 6].

3.3.1.3 Mesh

Mesh generation for the tank in this case followed the same meshing strategy as described for the tank in case 1, using the polyhedral mesher combined with prism layers and the surface remesher. The Two-Layer All y+ Wall Treatment was again applied for all wall boundaries. The number and placement of prism layers, as well as the treatment of the cylindrical interface between the rotating and stationary regions, remained unchanged. The maximum cell size and the target surface size on the impeller was adjusted to reach mesh independence. All other meshing parameters, including growth rate and boundary conditions, were kept consistent with the first case.

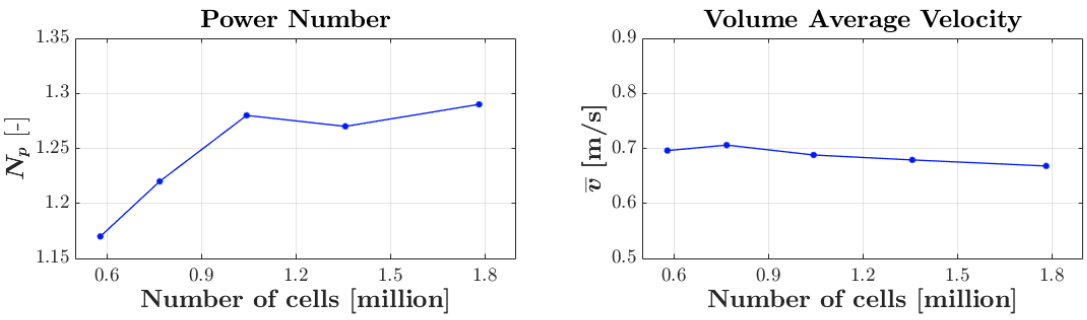
3.3.1.4 Mesh Independence Study

As in the previous case (see Section 3.1.1.4), only steady-state, single-phase simulations were performed at a constant impeller speed of 173 RPM.

Table 3.10: Mesh independence study for the FVM simulations in Case 3, including total cell count, maximum cell size, and target size on the impeller surface.

Number of cells [million]	**Maximum cell size [mm]**	**Impeller surface target size [mm]**
0.580	15	1
0.768	12	0.9
1.05	10	0.8
1.36	9	0.7
1.78	7	0.7

The refinement for this tank are summarized in Table 3.10, ranging from approximately 0.6 to 1.8 million cells. The cell count was increased progressively by modifying the maximum cell size and impeller surface target size, using a similar refinement factor as in the previous cases.

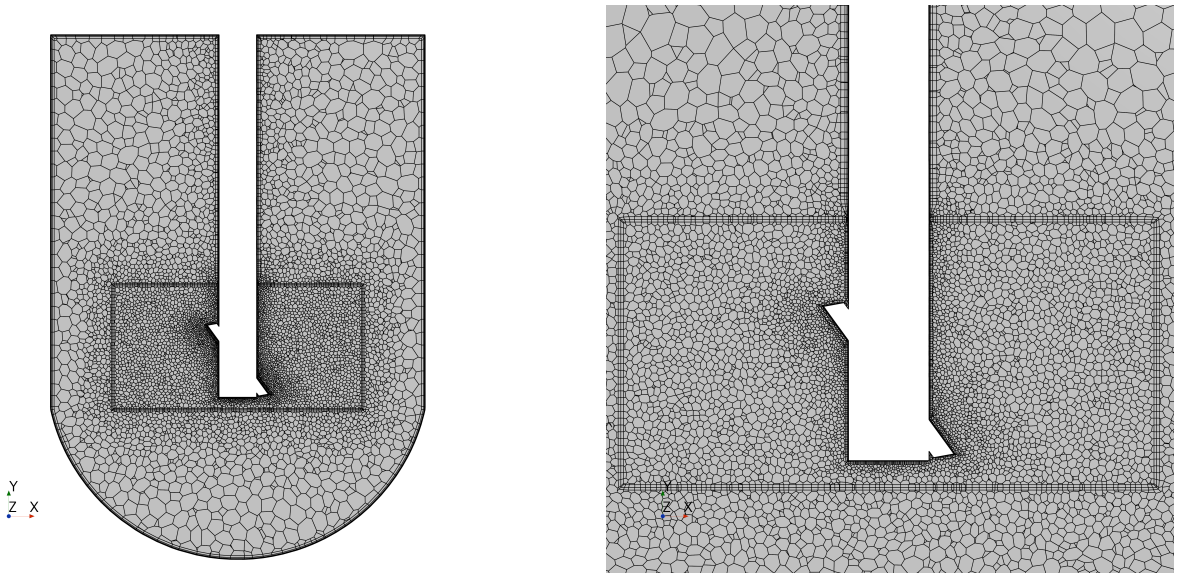


(a) Power number as a function of number of cells.

(b) Volume average velocity as a function of number of cells.

Figure 3.22: Mesh independence study for the FVM simulations in Case 3 with power number and volume average velocity analyzed as functions of the number of cells.

To evaluate mesh sensitivity, the same key parameters were monitored as in the sensitivity study of the MRF region. Figure 3.22 shows how these quantities vary with increasing mesh resolution. The results indicate that the solution stabilizes around 1.05 million cells. This is further supported by GCI estimates, which remained below 1% for both parameters at this resolution.



(a) Cross sectional mesh configuration in the XY-plane over the tank.

(b) Cross sectional mesh configuration in the XY-plane around the impeller.

Figure 3.23: Mesh configuration for the mesh used in Case 3.

Figure 3.23 displays the mesh used for solving the flow field.

3.3.1.5 Vortex Simulation Approach

The transient simulations were initiated from the steady-state solution obtained from the mesh independence study. The VOF solver was applied, together with a surface tension force between water and air of 0.072 N/m. In order to refine the mesh to sufficiently capture the vortex, the adaptive mesh solver was applied. An adaptive mesh criterion was set up based on the volume fraction of water. Regions where the water volume fraction was less than 0.2 were refined with a maximum refinement level of 3. The mesh was updated every 0.25 seconds until the vortex reached a stable shape and desired size.

Once the vortex had formed, the free surface was exported as a geometry part and excluded from the fluid domain in a similar way as in case 1. The adaptive mesh solver and VOF solver were then disabled, and the mesh was refined manually around the water surface using a target surface size of 50% which corresponds to 0.005 mm. A free-slip boundary condition was applied to the vortex surface. Figure 3.24 shows the final mesh used for the particle simulations.

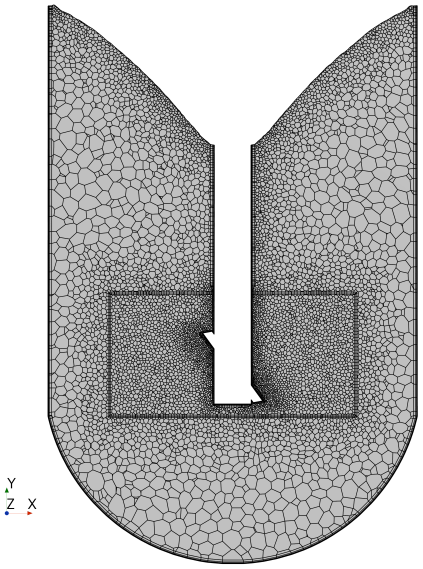


Figure 3.24: Cross sectional mesh configuration in the XY-plane used for the particle simulations after vortex formation in Case 3.

The simulation was then run for an additional 4 seconds to allow the flow field to stabilize before particle injection.

3.3.1.6 Evaluation of Particle Distributions

To simulate the particle phase using the FVM approach, the Lagrangian multiphase model was applied. The number of particles injected corresponded to a solid holdup of 0.005.

To evaluate how the particles were distributed in the tank, five probe arrays were placed similarly to the experimental setup. Each array contained six equally spaced probe points located at five different heights: 0.14, 0.20, 0.24, 0.28, and 0.35 meters from the bottom of the tank. The probe positions are illustrated in Figure 3.25.

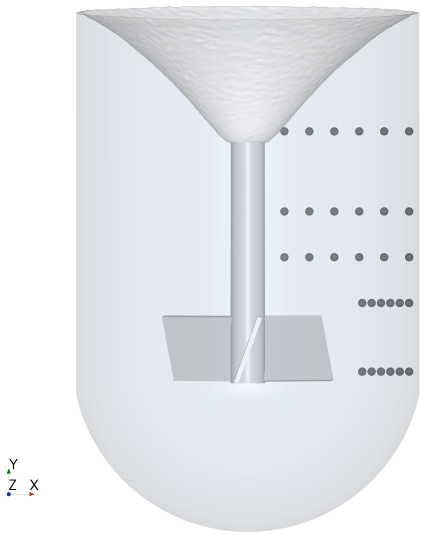


Figure 3.25: Location of probe arrays used to evaluate particle distribution in the biomedical tank configuration of Case 3. Each array contains six equally spaced probe points placed at five different heights in the y-direction: 0.14, 0.20, 0.24, 0.28, and 0.35 meters from the bottom of the tank.

Particles were injected just below the water surface over a time span of 0.1 second. The simulation then continued for approximately 1 minute to allow the particles to distribute within the fluid domain. The solid volume fraction was measured along the probe arrays shown in Figure 3.25 and averaged over the following 32 seconds, in order to match the experimental setup.

To reduce computational time, the use of particle parcels was considered, since the total number of individual particles was extremely large due to their small size. Parcels are groups of particles treated as single entities. However, this approach introduced challenges as a sufficient number of parcels was still required to achieve valid results. The large number of parcels leads to long simulation times. Unfortunately, there was not enough time within the project to resolve these issues and perform a complete evaluation using the FVM approach.

3.3.2 LBM Simulations - Case 3

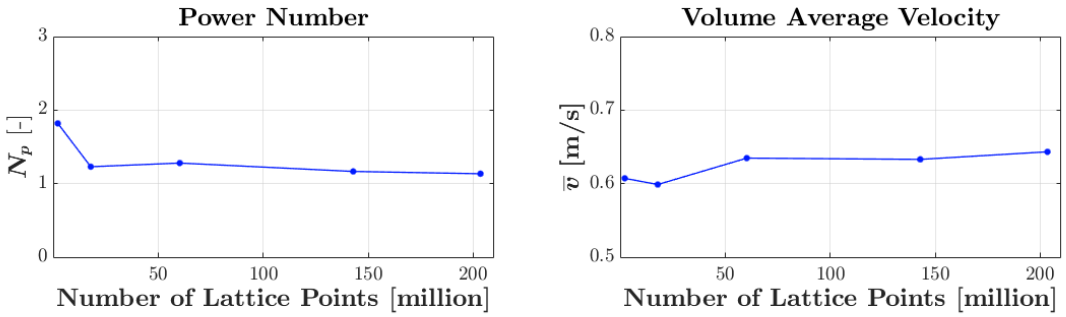
Below follows a brief description of the simulation method used for the LBM simulation of case 3. Similar to case 1 and 2, the setup was designed to resemble the FVM simulation as closely as possible.

3.3.2.1 Models

In the second biomedical tank, the free surface model is once again applied in the LBM solver. Similar to the biomedical tank 1, a three-dimensional transient solver was applied. LES was used as for turbulence model and VOF was applied for solving the interface between the air and water. A D3Q19 lattice was used. A surface tension force between air and water of 0.072 N/m was applied. The immersed boundary method was used in order to model the rotating body. Once the flow field and vortex had stabilized (about 20 seconds), the particles was injected. This was done by dumping the amount of particles which corresponds to a solid holdup of 0.005 in the tank. The inertial particle solver was used in LBM solver, which models particles with mass and movement according to Newton's second law, with uniform initial size distribution. The inertial particle momentum is modeled as one-way coupled to the fluid. Furthermore, injection downsampling was applied while using the inertial particle solver. Injection downsampling reduces the number of explicitly tracked particles by grouping many identical particles into parcels. Each parcel represents multiple particles but maintains the same diameter as a single particle. This approach lowers memory usage while preserving overall mass and interaction effects through internal scaling.

The simulation was first run with only gravity force and drag force activated, and then a second simulation were run where virtual mass and lift force were included in the simulation result. The additional forces in the simulation showed no significant impact on the measured particle distribution.

3.3.2.2 Resolution Independence Study



(a) Power number as a function of number of lattice points.

(b) Volume average velocity as a function of number of lattice points.

Figure 3.26: Resolution independence study for the LBM simulations in Case 3 with power number and volume average velocity analyzed as functions of the number of lattice points.

A resolution independence study was performed for Case 3. According to the M-Star documentation, an accurate solution typically requires between 80 and 100 lattice points per impeller diameter [5]. In this study, resolutions ranging from 100 to 500 lattice points across the tank diameter were tested. A resolution of 200 lattice points across the tank, corresponding to 100 lattice points across the impeller, was found to be sufficient. This is in line with the documentation and resulted in a total of approximately 18 million lattice points.

The resolution was considered sufficient primarily based on the convergence of the power number, which showed minimal variation beyond this resolution. Figure 3.26 presents the results of the resolution study, where both power number and volume average velocity were evaluated.

4

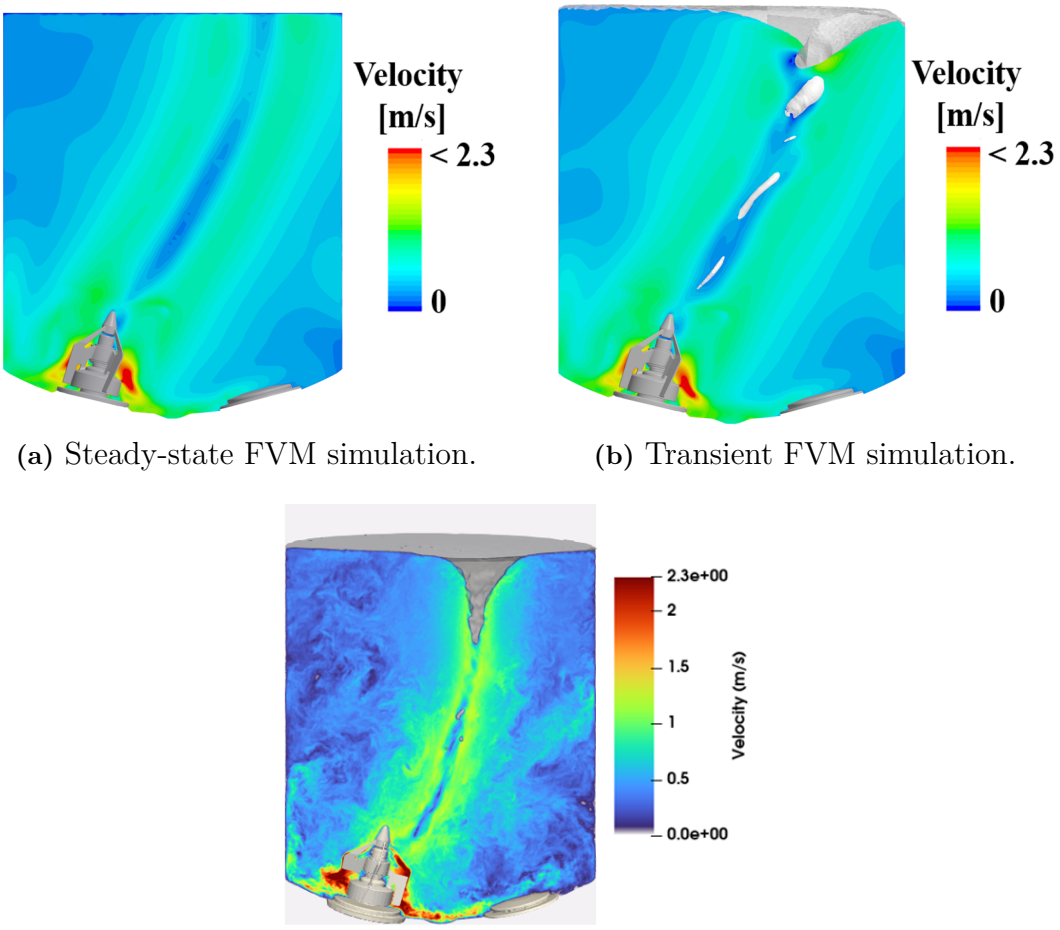
Results

This chapter presents the relevant results from the simulations performed, as described in the methodology, for each case. For Case 1 and Case 2, the flow field and impeller-related results are shown first, followed by the mixing time and solver performance. For Case 3, flow and impeller results are presented first, followed by particle distribution results.

4.1 Case 1

This section presents the simulation results for case 1 using both the FVM and LBM methods. The results include analysis of the flow field and vortex formation, comparison of impeller parameters, evaluation of mixing time and comparison of computational efficiency based on solver time.

4.1.1 Flow Results



(a) Steady-state FVM simulation.

(b) Transient FVM simulation.

(c) LBM simulation.

Figure 4.1: Cross sectional velocity field for the FVM steady-state, FVM transient, and LBM simulations at an impeller speed of 337 RPM. Vortex formation is visible in both the transient FVM and LBM flow fields.

Figure 4.1 shows the velocity field for the three simulations at 337 RPM. The steady FVM simulation gives a smooth and symmetric flow, without unsteady structures. The transient FVM simulation captures vortex formation and asymmetry in the jet region. The LBM simulation shows a more detailed flow, with stronger velocity gradients and smaller flow structures across the tank.

(a) Transient FVM simulation. **(b)** LBM simulation.

Figure 4.2: Vortex shape captured in the FVM transient and LBM simulations at an impeller speed of 337 RPM.

Figure 4.2 shows the vortex shape in the FVM and LBM simulations at 337 RPM. The FVM vortex is wider and longer. The LBM vortex is narrower and shifts slightly sideways over time. This lateral motion is not captured in the FVM simulation, where the vortex stays mostly fixed.

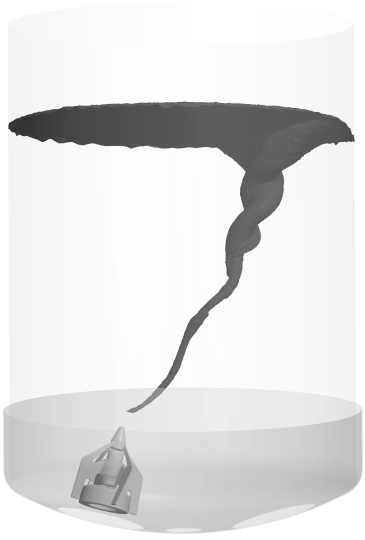
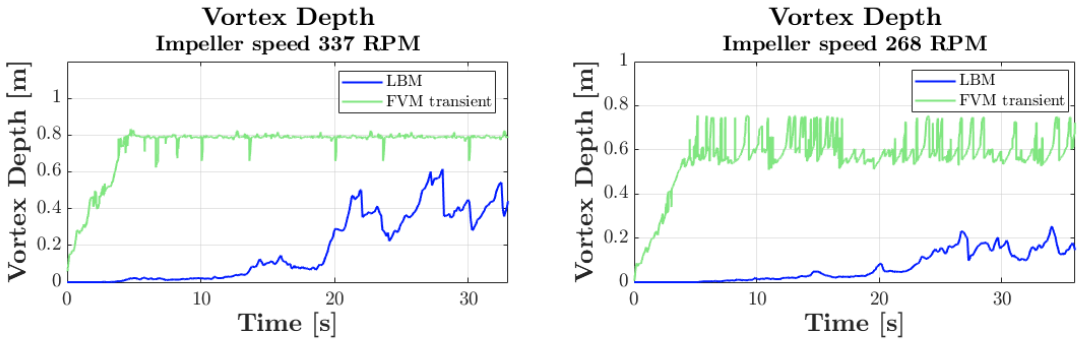


Figure 4.3: Time-averaged vortex surface from the transient FVM simulation at 337 RPM.

Figure 4.3 shows the average vortex from the FVM simulation at 337 RPM. Since the vortex fluctuates during the simulation, a time average was used to get a more

representative shape. The averaging also made it possible to analyze mixing without simulating the full transient vortex, which would have required much longer simulation times. In contrast, the LBM solver is significantly faster and has no difficulty resolving the transient vortex while simultaneously solving the passive scalar, so averaging was not necessary in that case.

Corresponding vortex visualizations for the impeller speed of 268 RPM are provided in Appendix A.0.1.



(a) Vortex depth over time for FVM and LBM simulations at 337 RPM.

(b) Vortex depth over time for FVM and LBM simulations at 268 RPM.

Figure 4.4: Vortex depth development in FVM and LBM simulations at two impeller speeds.

Figure 4.4 shows the vortex depth over time for both simulation methods. At 337 RPM, the FVM simulation predicts a deeper vortex than LBM. At 268 RPM, FVM reaches about 0.6–0.7 m quickly and stays at that level, while LBM builds up more slowly and stays below a value of 0.25 m. In FVM, fluctuations in vortex depth are present throughout the simulation but occur in a relatively consistent pattern. In contrast, LBM shows large variations in vortex depth that appear more irregular and less predictable. Based on experimental results, LBM clearly underestimate the vortex depth.

4.1.2 Impeller Parameters

For the impeller speed of 337 RPM, the power number results shown in Table 4.1 indicate that all three simulations predict the power number to be within the uncertainty of the experiment. However, all simulations result in a power number that falls within the upper range of the margin of error.

For the pumping number at 337 RPM, no experimental data is available. The steady and transient simulations of FVM result in slightly higher values compared to LBM. However, the pumping number predicted in the transient FVM simulation and the LBM simulation shows an almost negligible difference.

Table 4.1: Impeller parameters for the impeller speed of 337 RPM from the different simulation methods. Experimental reference data from Larsson and Lindenbaum [2].

	FVM Steady	**FVM Transient**	**LBM**	**Experiment**
Power Number [-]	3.86	3.80	3.75	3.6±0.4
Pumping Number [-]	0.654	0.646	0.639	-

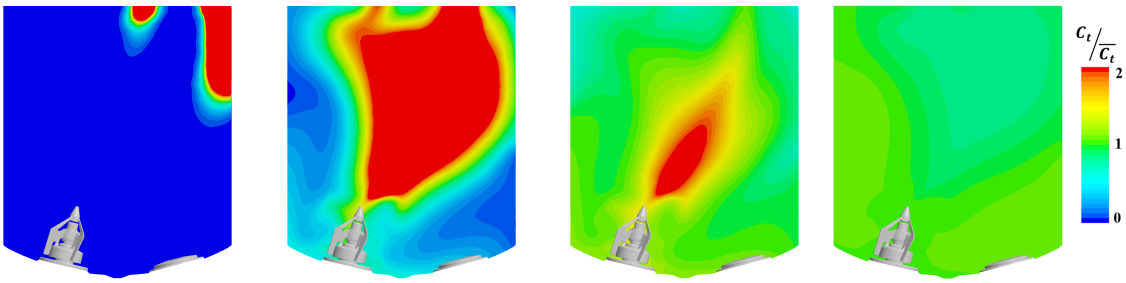
At the lower impeller speed of 268 RPM, the results in Table 4.2 show that all three simulation methods predict values close to the experimental power number. Both of the FVM simulations predict a power number of 3.89, and the LBM simulation predicts a slightly higher power number. However, as for the higher impeller speed, all simulations are well within the range of the uncertainty of the experiment. Regarding the pumping number, the difference between the result from the different simulations are almost negligible.

Table 4.2: Impeller parameters for the impeller speed of 268 RPM from the different simulation methods. Experimental reference data from Larsson and Lindenbaum [2].

	FVM Steady	**FVM Transient**	**LBM**	**Experiment**
Power Number [-]	3.89	3.89	3.98	3.8±0.5
Pumping Number [-]	0.654	0.658	0.656	-

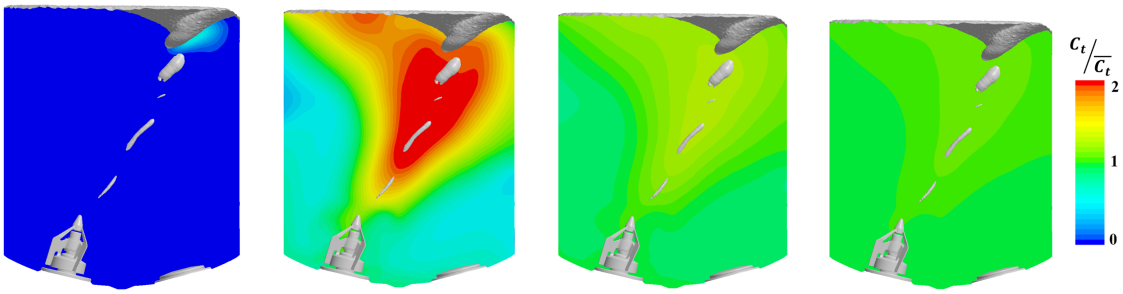
4.1.3 Mixing time

The mixing process is illustrated in the contour plots of normalized concentration shown in Figures 4.5, 4.6, and 4.7, which depicts how the tracer spreads throughout the domain over time. These visualizations confirm that the transient methods predict a faster and more realistic mixing behavior compared to the steady-state approach. The same behavior can be observed for the lower impeller speed.



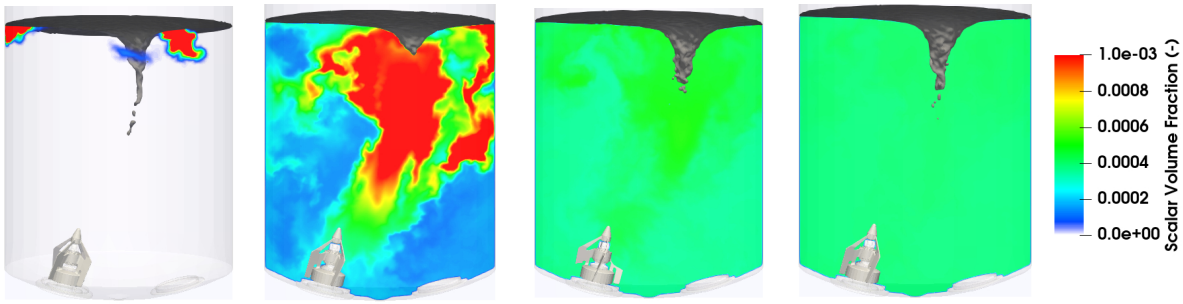
(a) t=2 s **(b)** t=10 s **(c)** t=18 s **(d)** t=25 s

Figure 4.5: Contour plots of normalized concentration over time, FVM steady simulation.



(a) t=2 s (b) t=10 s (c) t=18 s (d) t=25 s

Figure 4.6: Contour plots of normalized concentration over time, FVM transient simulation.



(a) t=2 s (b) t=10 s (c) t=18 s (d) t=25 s

Figure 4.7: Contour plots of scalar volume fraction over time, LBM simulation.

For the impeller speed of 337 RPM, the results presented in Table 4.3, show that the steady-state FVM simulation overpredicts the mixing time by approximately 32% compared to the experimental data. In contrast, the transient FVM and LBM simulations result in much smaller deviations. The mixing times predicted by the transient FVM and LBM simulations are within the experimental uncertainty, regardless of the measurement method.

Table 4.3: Mixing time for the impeller speed of 337 RPM from the different simulation methods. Experimental reference data from Larsson and Lindenbaum [2].

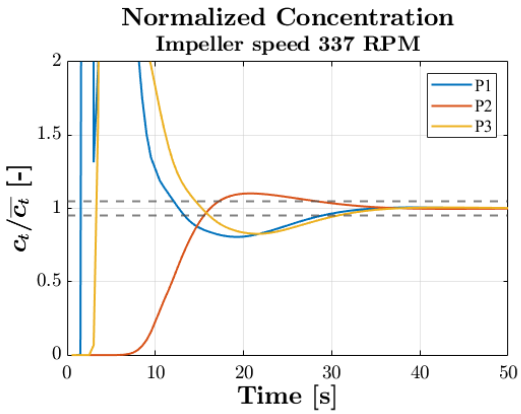
	FVM Steady	**FVM Transient**	**LBM**	**Experiment**
Mixing time probes [s]	33.0	26.7	26.0	25±3
Mixing time CV [s]	33.0	26.5	25.6	-

At the lower impeller speed of 268 RPM, the steady-state FVM simulation shows the largest deviation from the experimental probe-based measurements, overpredicting

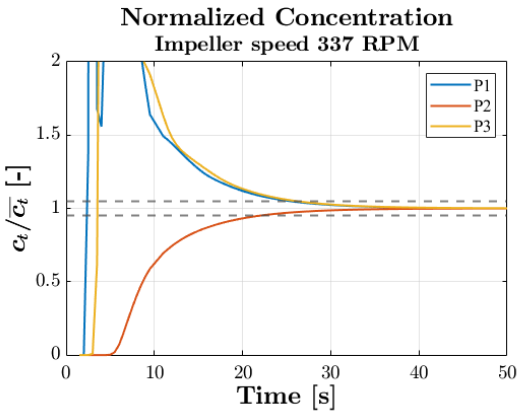
the mixing time by about 17%, as shown in Table 4.4. The transient FVM simulation predicts a slightly lower mixing time than the LBM simulation, but both simulations still predict a mixing time within the margin of error of the experiment. The mixing time estimated by CV follows a similar trend.

Table 4.4: Mixing time for the impeller speed of 268 RPM from the different simulation methods. Experimental reference data from Larsson and Lindenbaum [2].

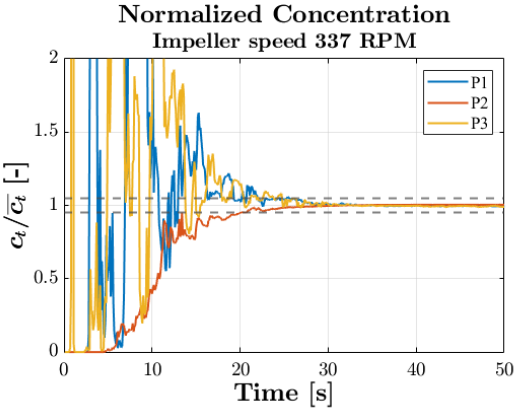
	FVM Steady	**FVM Transient**	**LBM**	**Experiment**
Mixing time probes [s]	35.0	28.5	30.4	30±4
Mixing time CV [s]	37.4	30.0	31.8	-



(a) Normalized tracer concentration at three probe points from the steady-state FVM simulation.



(b) Normalized tracer concentration from the transient FVM simulation.



(c) Normalized tracer concentration from the LBM simulation.

Figure 4.8: Time evolution of normalized tracer concentration at three probe locations for the different simulation approaches at 337 RPM. Mixing is considered complete when the normalized concentration at all three probes reaches and remains within the interval 0.95–1.05. This interval is indicated by grey dashed horizontal lines in all plots.

Figure 4.8 shows the evolution in time of the normalized concentration at three probe locations for an impeller speed of 337 RPM. When comparing steady-state FVM, transient FVM, and LBM simulations, it is evident that steady-state FVM significantly overpredicts mixing time. The corresponding concentration contour plots and probe graphs for all three simulation methods at 268 RPM are provided in Appendix A.0.2 and A.0.3.

4.1.4 Solver Time

Table 4.5 shows both the CPU/GPU time in hours and the normalized CPU/GPU time for all simulation methods. The CPU/GPU time refers to the physical time required to complete the simulations. The FVM simulations were run on 64 CPU cores, while the LBM simulations were executed on the GPU. The normalized CPU/GPU time is calculated by dividing the CPU/GPU time by the number of simulated physical minutes. This allows for a fairer comparison between simulations with different durations. For both steady-state FVM simulations and transient FVM simulations, the time required to obtain a converged flow field in the initial steady-state run is included in the total solver time. However, this initialization phase does not generate any simulated time and is therefore still reflected in the normalized solver time.

Table 4.5: Solver time results for the different simulation approaches at 337 RPM, including total CPU/GPU time and time normalized by simulated physical minutes.

	FVM Steady	**FVM Transient**	**LBM**
CPU/GPU time [h]	9.4	251	54.6
Normalized CPU/GPU time [h]	11.3	182	27.3

The LBM simulation for the impeller speed of 337 RPM was approximately 4.6 times faster than the transient FVM simulation in terms of total solver time and 6.7 times more efficient when normalized per simulated minute. Of all simulation methods, the steady FVM simulation was the most computationally efficient, which was expected.

Table 4.6: Solver time results for the different simulation approaches at 268 RPM, including total CPU/GPU time and time normalized by simulated physical minutes.

	FVM Steady	**FVM Transient**	**LBM**
CPU/GPU time [h]	6.76	472	51.3
Normalized CPU/GPU time [h]	8.11	329	25.7

The LBM simulation for the impeller speed of 268 RPM was approximately 9.2 times faster than the transient FVM simulation in terms of total solver time, and 12.8 times more efficient when normalized per simulated minute. These results are summarized in Table 4.6. Also, for this case, the steady FVM simulation had a significantly lower CPU time.

4.2 Case 2

This section presents the results from both the FVM and LBM simulations for the digester case, including flow field visualizations and mixing time analyses.

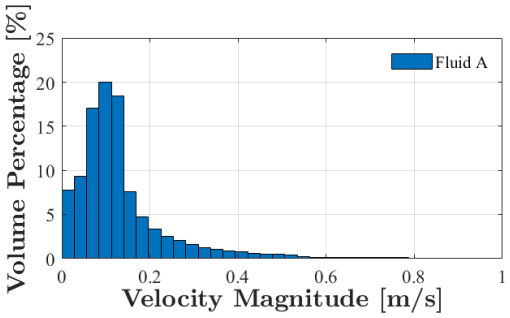
4.2.1 Flow Results and Impeller Parameters

The key tank and impeller parameters from the final FVM and LBM simulations using configuration r1, with a resolution of 200 and refinement level 2, are presented in Table 4.7. For all results, the LBM simulations tended to predict higher values for impeller-related parameters, while global tank parameters such as mean velocity and viscosity were generally lower compared to the FVM results.

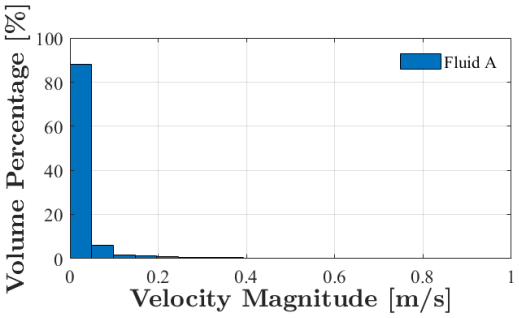
Table 4.7: Key parameters for fluid A and fluid B measured in the FVM and LBM simulations with configuration r1.

	Fluid A		**Fluid B**	
	FVM	**LBM**	**FVM**	**LBM**
$\overline{v}$ [m/s]	0.141	0.0321	0.0876	0.0232
$\overline{\mu}$ [Pa s]	47.7	496	145	698
N_{P1} [-]	0.528	1.19	0.517	1.19
N_{P2} [-]	0.410	1.23	0.407	1.201
T_1 [Nm]	367	810	360	817
T_2 [Nm]	285	841	283	835

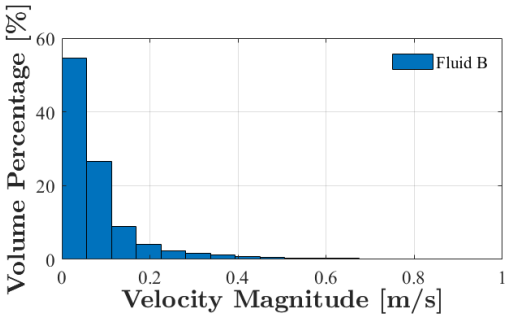
Histograms showing the velocity and viscosity distributions in the tank, expressed as volume percentages, are presented in Figure 4.9 and 4.10 respectively, comparing the results from the FVM and LBM simulations using configuration r1.



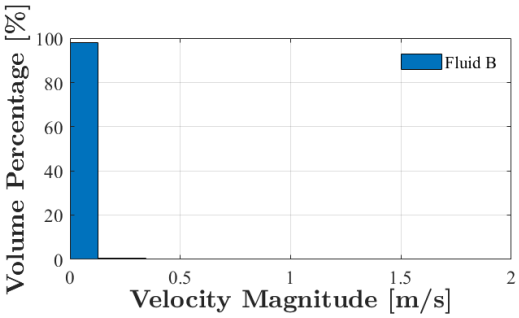
(a) Velocity distribution for fluid A using FVM.



(b) Velocity distribution for fluid A using LBM.



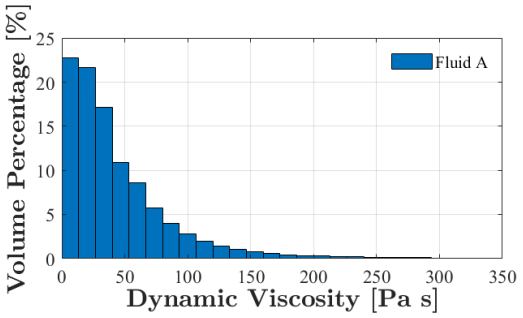
(c) Velocity distribution for fluid B using FVM.



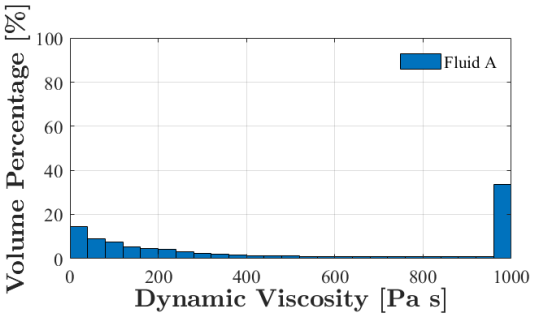
(d) Velocity distribution for fluid B using LBM.

Figure 4.9: Histogram for velocity distribution over volume percentage of the tank for fluid A and B comparing FVM and LBM.

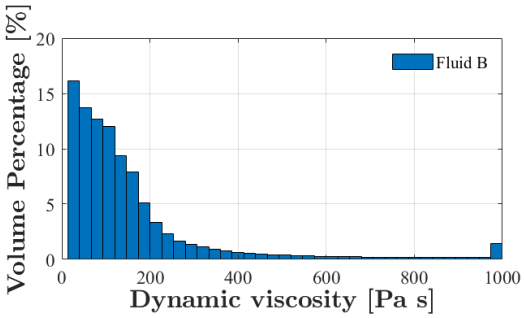
The FVM simulations exhibit a more uniform velocity distribution compared to the LBM simulations for both fluids. Although the mean velocities in the LBM simulations are lower, a significant portion of the tank experiences velocities close to zero. In the LBM simulations with fluid B, almost 100% of the tank is stagnant.



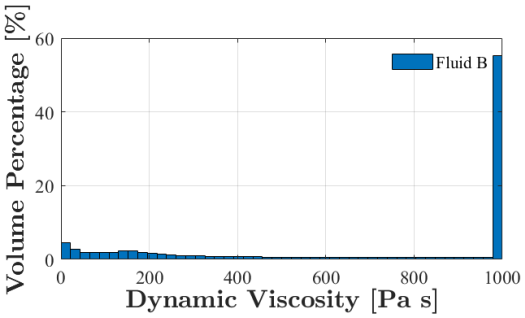
(a) Dynamic viscosity distribution for fluid A using FVM.



(b) Dynamic viscosity distribution for fluid A using LBM.



(c) Dynamic viscosity distribution for fluid B using FVM.

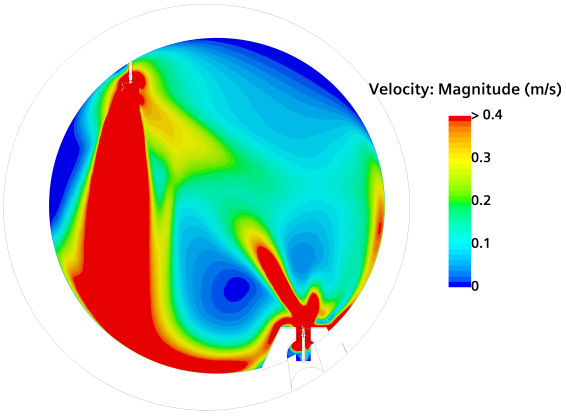


(d) Dynamic viscosity distribution for fluid B using LBM.

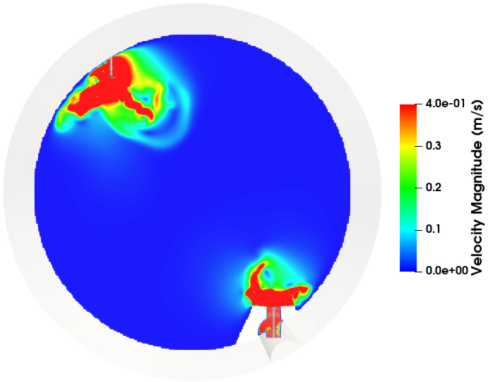
Figure 4.10: Histogram for dynamic viscosity distribution over volume percentage of the tank for fluid A and B comparing FVM and LBM.

Similar to the velocity distribution, and due to the strong dependence of velocity on viscosity, the FVM simulation exhibits a more uniform viscosity distribution. In contrast, the LBM simulation shows a significant portion of the tank with viscosity values at the maximum limit for the viscosity at 1000 Pa·s.

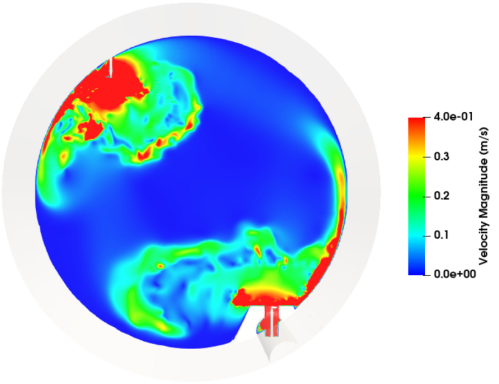
Velocity and viscosity contour plots comparing the FVM and LBM simulations are shown in Figure 4.11 and 4.12. The results from the additional simulations with configuration r2 and the uniform lattice are also included in these figures for comparison. The r2 configuration has a resolution of 200 and refinement level 3, while the uniform lattice configurations has a resolution of 700 lattice points.



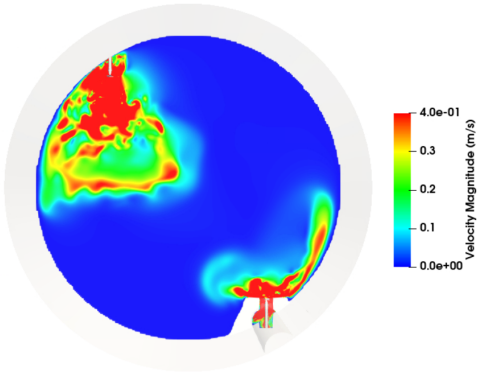
(a) Velocity contour plots for FVM simulation.



(b) Velocity contour plots for LBM simulation with configuration r1.



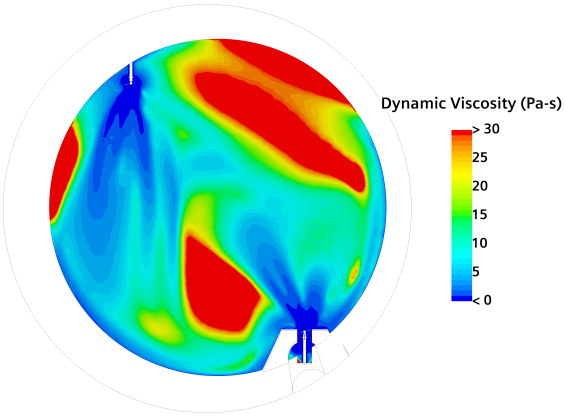
(c) Velocity contour plots for LBM simulation with the uniform lattice.



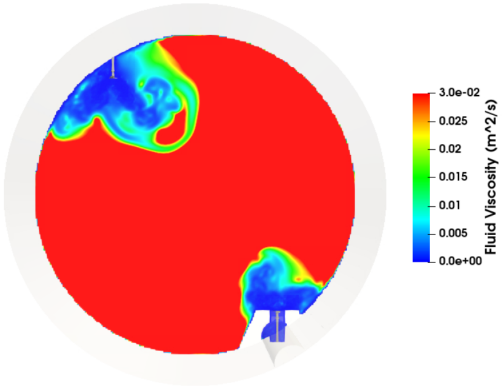
(d) Velocity contour plots for LBM simulation with configuration r2.

Figure 4.11: Velocity magnitude contour plots up to 0.4 m/s comparing FVM with LBM for three different configurations.

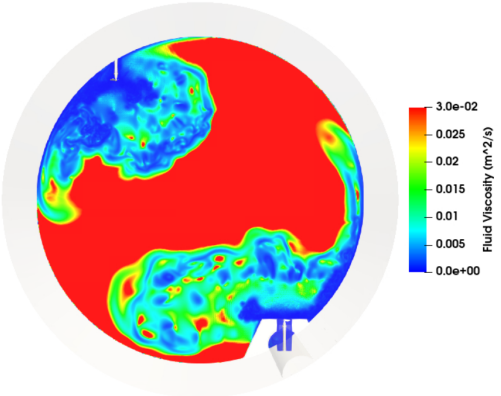
Note that in Figure 4.12 the viscosity shown is the kinematic viscosity, not dynamic viscosity. The unit is in m^2/s and can be converted to Pa·s by multiplying the kinematic viscosity by the density ($1000kg/m^3$). All figures show the same scale for the fluid viscosity.



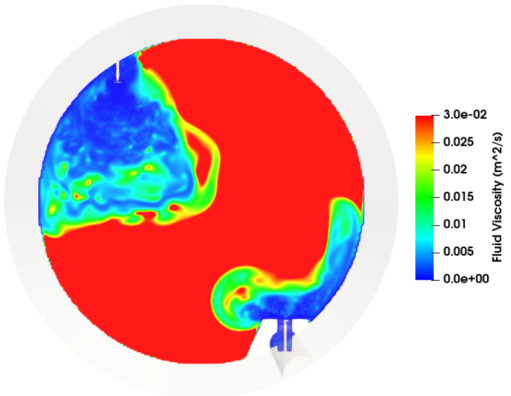
(a) Viscosity contour plots for FVM simulation.



(b) Viscosity contour plots for LBM simulation with configuration r1.



(c) Viscosity contour plots for LBM simulation with the uniform lattice.

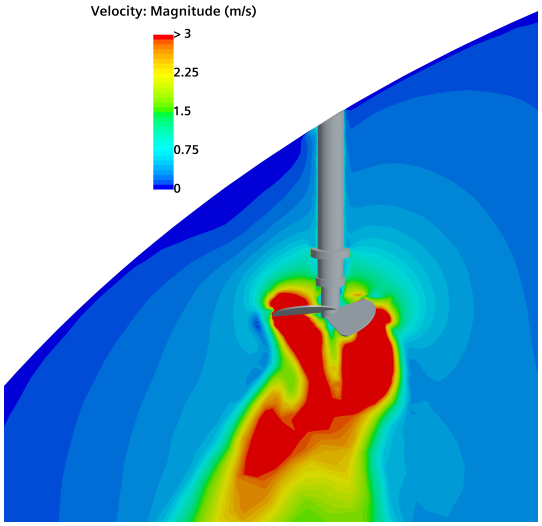


(d) Viscosity contour plots for LBM simulation with configuration r2.

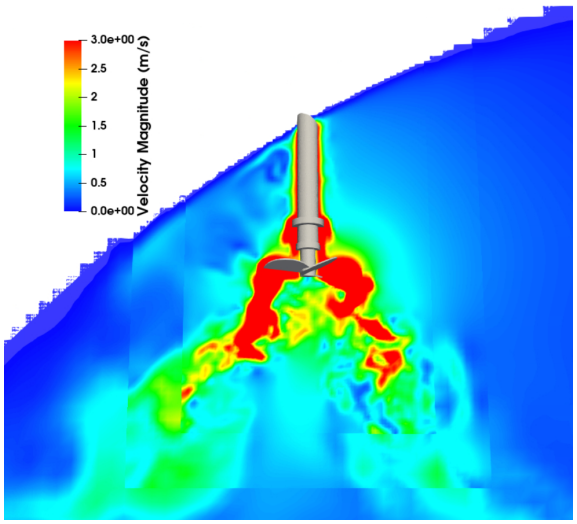
Figure 4.12: Viscosity magnitude contour plots up to 30 Pa·s comparing FVM with LBM for three different configurations.

According to the contour plots, none of the LBM simulations gave a fully developed flow field as seen in the FVM simulations. The simulation with configuration r2 gave a more developed flow field than configuration r1. The uniform lattice simulation gave the most developed flow field, especially around impeller 2. This can also be seen in Table 4.7 since this configuration experiences the highest velocity and lowest dynamic viscosity.

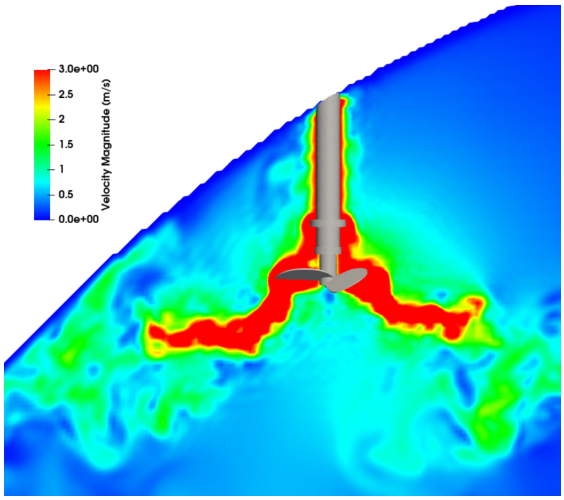
Figure 4.13 and 4.14 exhibit a velocity contour plot comparison between the FVM and LBM simulations for impeller 1 and impeller 2 respectively.



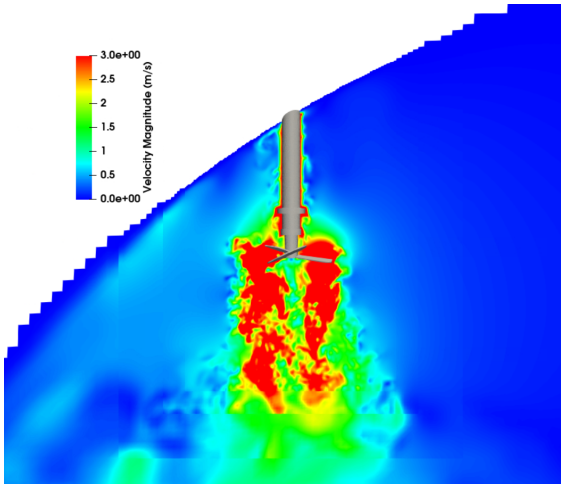
(a) Velocity contour plot for impeller 1 using FVM.



(b) Velocity contour plot for impeller 1 using LBM configuration r1.

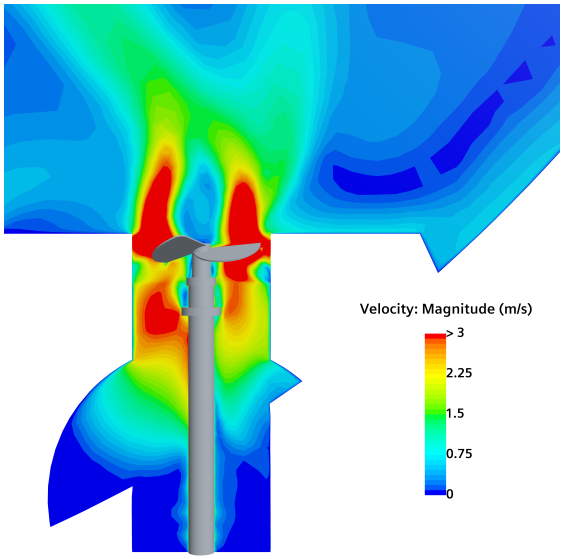


(c) Velocity contour plot for impeller 1 using LBM with the uniform lattice.

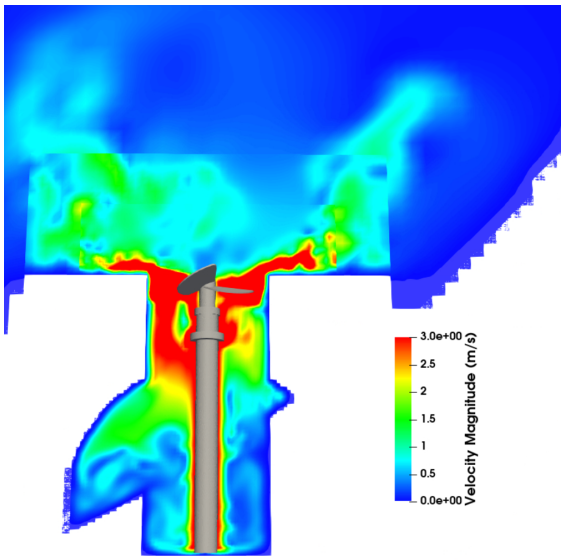


(d) Velocity contour plot for impeller 1 using LBM configuration r2.

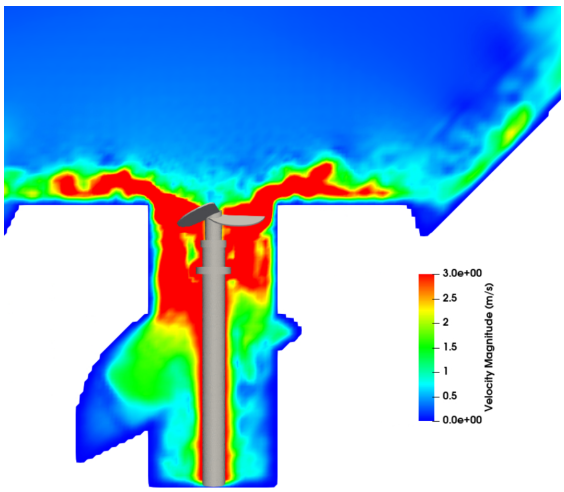
Figure 4.13: Velocity contour up to 3 m/s comparing the flow field around impeller 1 between FVM and three different configuratinos using LBM.



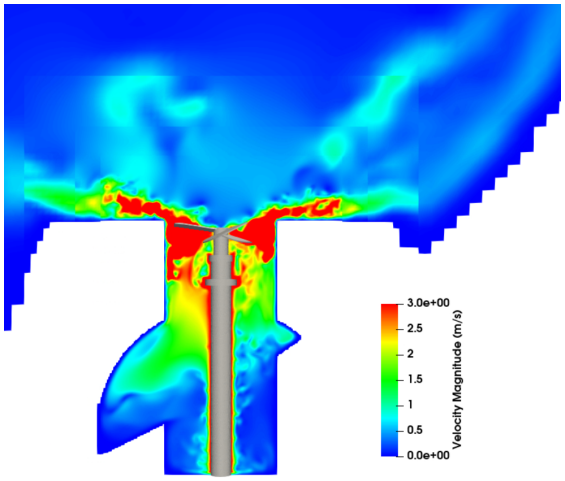
(a) Velocity contour plot for impeller 2 using FVM.



(b) Velocity contour plot for impeller 2 using LBM configuration r1.



(c) Velocity contour plot for impeller 2 using LBM with the uniform lattice.



(d) Velocity contour plot for impeller 2 using LBM configuration r2.

Figure 4.14: Velocity contour up to 3 m/s comparing the flow field around impeller 2 between FVM and three different configuratinos using LBM.

The results for impeller 1 demonstrate a dependency of the flow field on the spatial resolution around the impeller. As the spatial resolution is refined and dx_r decreases, the simulation captures more physically realistic features of the flow. In contrast, impeller 2 does not show any notable improvement in flow features with decreasing dx_r. Regardless of the grid configuration, the velocity distribution remains physically unrealistic with a dominant radial velocity. This suggests that even with increased local resolution, the simulation fails to capture the correct flow dynamics around impeller 2.

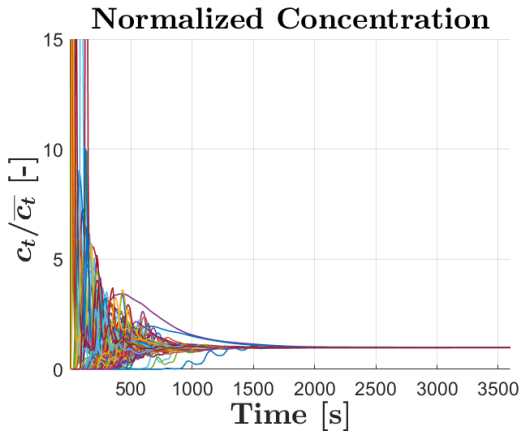
Table 4.8 presents the key impeller and tank parameters for all additional LBM simulations and the FVM simulation. The results show that the solution is not time step size independent, and by reducing the time step size, the difference in values between the FVM and LBM simulations increases. The uniform lattice setup gave the highest mean velocity among the LBM simulations. However, it also showed the biggest difference compared to the FVM simulations in terms of impeller parameters, where r1, and especially r2, were more closely aligned with the FVM results.

Table 4.8: Results comparing the simulations with fluid A comparing FVM with LBM for all additional simulations.

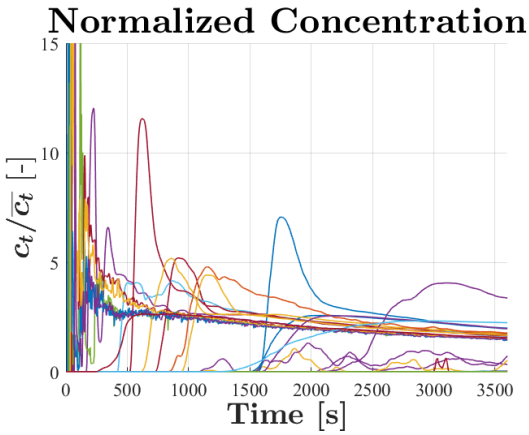
	FVM	**LBM: r1**	**LBM: r1, Co=0.05**	**LBM: 700**	**LBM: r2**
$\overline{v}$ [m/s]	0.141	0.0321	0.0247	0.0558	0.0291
$\overline{\mu}$ [Pa s]	47.7	496	597	273	554
N_{P1} [-]	0.528	1.19	1.27	1.48	0.608
N_{P2} [-]	0.410	1.23	1.39	1.66	0.767
T_1 [Nm]	367	810	869	999	415
T_2 [Nm]	285	841	944	1130	520

4.2.2 Mixing Time

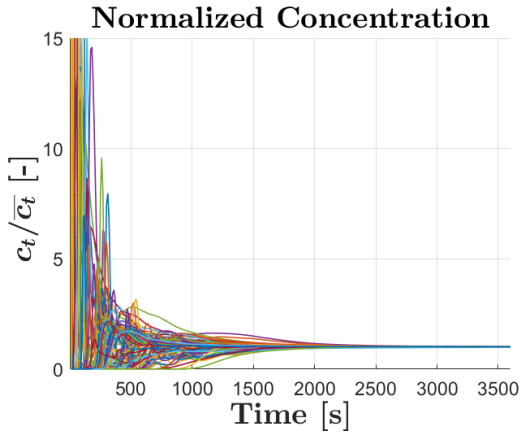
Figure 4.15 illustrates the mixing process in the tank by showing the normalized tracer concentration, where a value of 1 indicates perfect mixing, measured at 85 probe points distributed throughout the tank. The mixing time is presented in Table 4.9. In the FVM simulations, the mixing time for fluid B was approximately 15 minutes longer than for fluid A. In contrast, none of the LBM simulations reached the mixing criterion within 3600s. Several probe points still displayed values near zero, indicating that the tracer had not yet reached those locations. Only a small number of probe points were within the required 5% range.



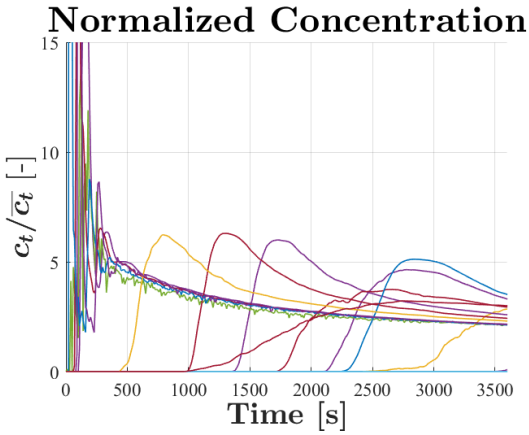
(a) Normalized concentration of tracer over time for fluid A using FVM.



(b) Normalized concentration of tracer over time for fluid A using LBM.



(c) Normalized concentration of tracer over time for fluid B using FVM.



(d) Normalized concentration of tracer over time for fluid B using LBM.

Figure 4.15: Normalized concentration over time showing the mixing procedure over 85 different probes comparing fluid A and B with FVM and LBM simulations.

Table 4.9: Mixing time for FVM and LBM simulations for both fluids.

Mixing time - FVM		Mixing time - LBM	
Fluid A	**Fluid B**	**Fluid A**	**Fluid B**
1800 s	2700 s	-	-

4.2.3 Solver Time

The solver time for the mixing time simulations is presented in Table 4.11. For the FVM simulation, the solver time includes both the steady simulations to converge the flow field and the passive scalar simulations on a frozen field. The LBM simulation includes the transient simulations for both the flow field to converge and the passive scalar. Since the LBM did not reach the mixing criterion, the solver time in Table 4.10 is based on a 3600 s long simulation time, to provide a fair comparison.

Table 4.10: CPU/GPU time for mixing time FVM and LBM simulations.

	Fluid A		**Fluid B**	
	FVM	**LBM**	**FVM**	**LBM**
CPU/GPU time [h]	16.3	28.96	10	25.32
Normalized CPU/GPU time [h]	0.273	0.434	0.167	0.380

The results show that the LBM simulations have a higher solver time, both for the entire simulation and the normalized value. The LBM simulations require a simulation time up to 2.53 times longer than the FVM simulation for the total CPU time and up to 2.28 times longer per simulation minute.

Table 4.11 presents the total and normalized GPU time for all additional simulations using LBM until a converged flow field is reached. The results show that the simulation time increases significantly when using a uniform lattice, requiring approximately 17 times longer simulation time compared to the chosen configuration. Maintaining the main resolution but increasing the refinement level from two to three (configuration r1 and r2) results in a simulation time around 4.5 times longer.

Table 4.11: GPU time for all configurations using LBM with estimated time for mixing time simulation.

LBM configuration:	**r1**	**r1, Co=0.05**	**700**	**r2**
GPU time [h]	1.81	2.89	15.85	7.8
Normalized GPU time [h]	0.434	0.69	7.934	1.95
Estimated time for simulating mixing time [h]	28.96	46.3	493	130

4.3 Case 3

This section presents the simulation results for Case 3 using both the FVM and LBM methods. The results include analysis of the flow field and vortex formation, comparison of impeller parameters, evaluation of particle distribution, and assessment of computational performance.

No final results could be obtained for the transient FVM simulations due to long computation times and the inability to reach meaningful outcomes within the time frame of this project. However, available FVM results, such as the steady-state solution and the VOF-based approach used to capture the vortex, are still included in order to support the evaluation and validation of the LBM simulation.

4.3.1 Flow and Impeller Parameters Results

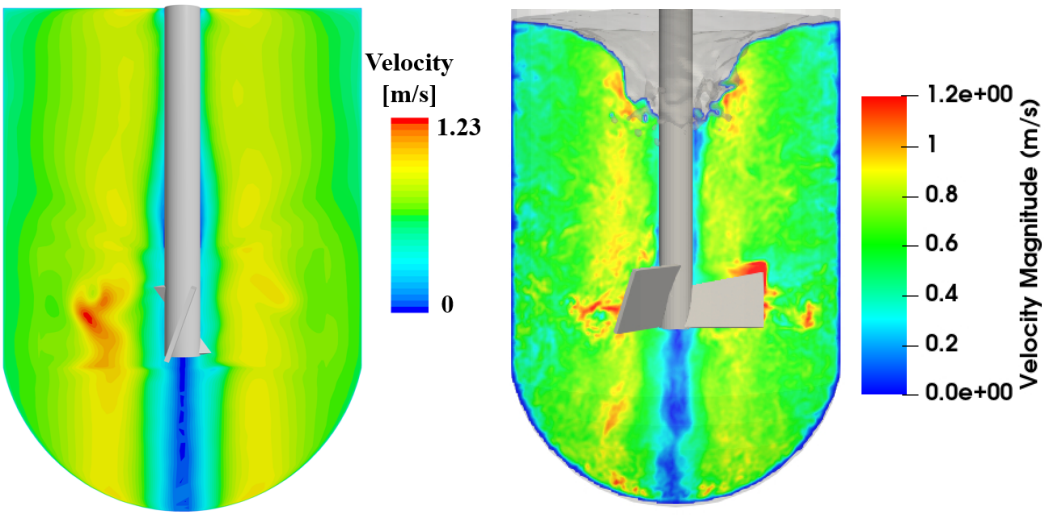
Table 4.12 presents the calculated impeller power number at 173 RPM for the different simulation methods. The results show good agreement between the steady-state FVM and the LBM simulations, with power numbers of 1.28 and 1.23, respectively.

No experimental data for the power number in this specific configuration is available for direct comparison.

Table 4.12: Impeller power number for the impeller speed of 173 RPM from the different simulation methods.

	FVM Steady	**FVM Transient**	**LBM**
Power Number [-]	1.28	-	1.23

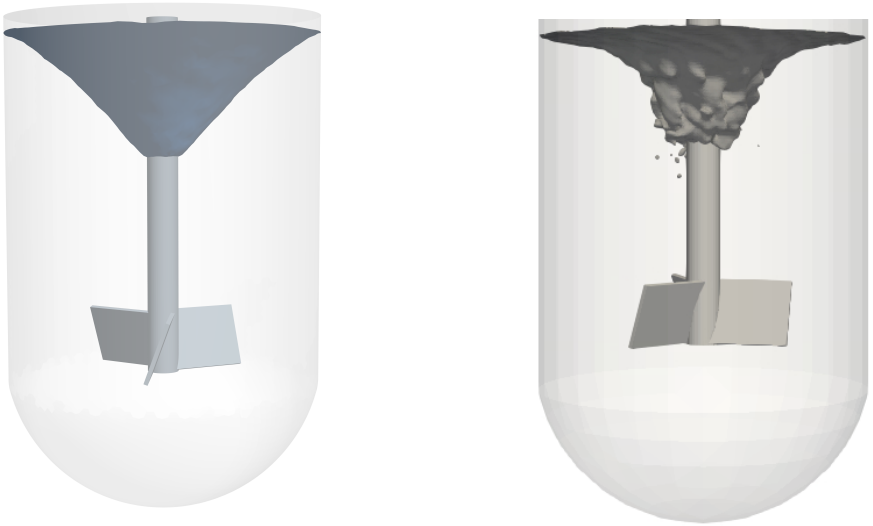
Figure 4.16 shows the cross-sectional velocity fields for the steady FVM and LBM simulations. While the steady FVM simulation cannot capture transient features such as the vortex, the overall structure of the velocity field is similar between the two methods. In both cases, a region of low velocity is observed directly below the impeller, consistent with expected flow patterns in stirred tanks.



(a) Steady FVM simulation. **(b)** LBM simulation.

Figure 4.16: Contour plots of the cross-sectional velocity field for the FVM steady-state and LBM simulations at an impeller speed of 173 RPM. Vortex formation is visible in the LBM flow field.

In Figure 4.17, the vortex shape is shown for the transient FVM and LBM simulations. A similar trend as in Case 1 is observed, where the FVM predicts a larger vortex compared to the LBM. Additionally, the vortex captured by the LBM appears less smooth and exhibits more temporal variation, which is likely more representative of the true unsteady behavior.



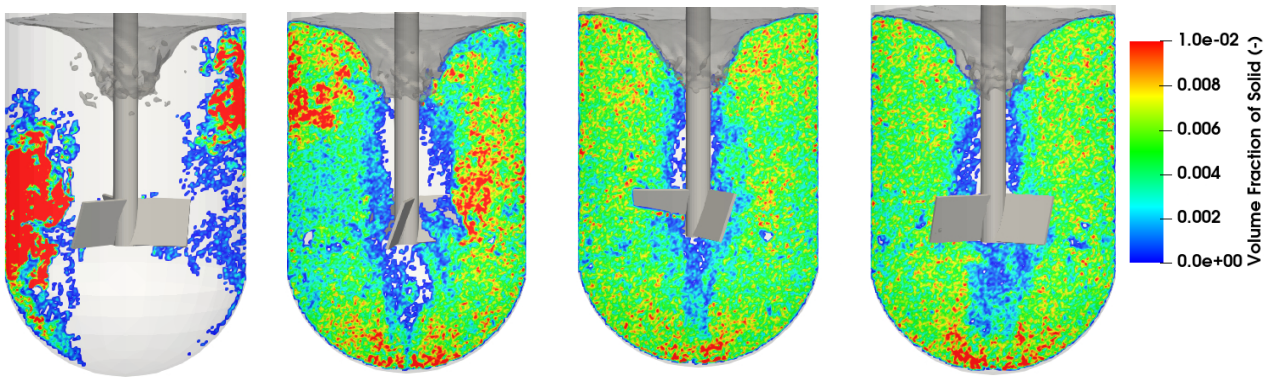
(a) Transient FVM simulation **(b)** LBM simulation

Figure 4.17: Vortex shape captured in the transient FVM and LBM simulations in Case 3 at an impeller speed of 173 RPM.

4.3.2 Particle Distribution

Figure 4.18 shows contour plots of the solid volume fraction ϕ over time from the LBM simulations. The time 0 s corresponds to the moment of particle injection into the tank, at which point the flow field has already reached a developed state.

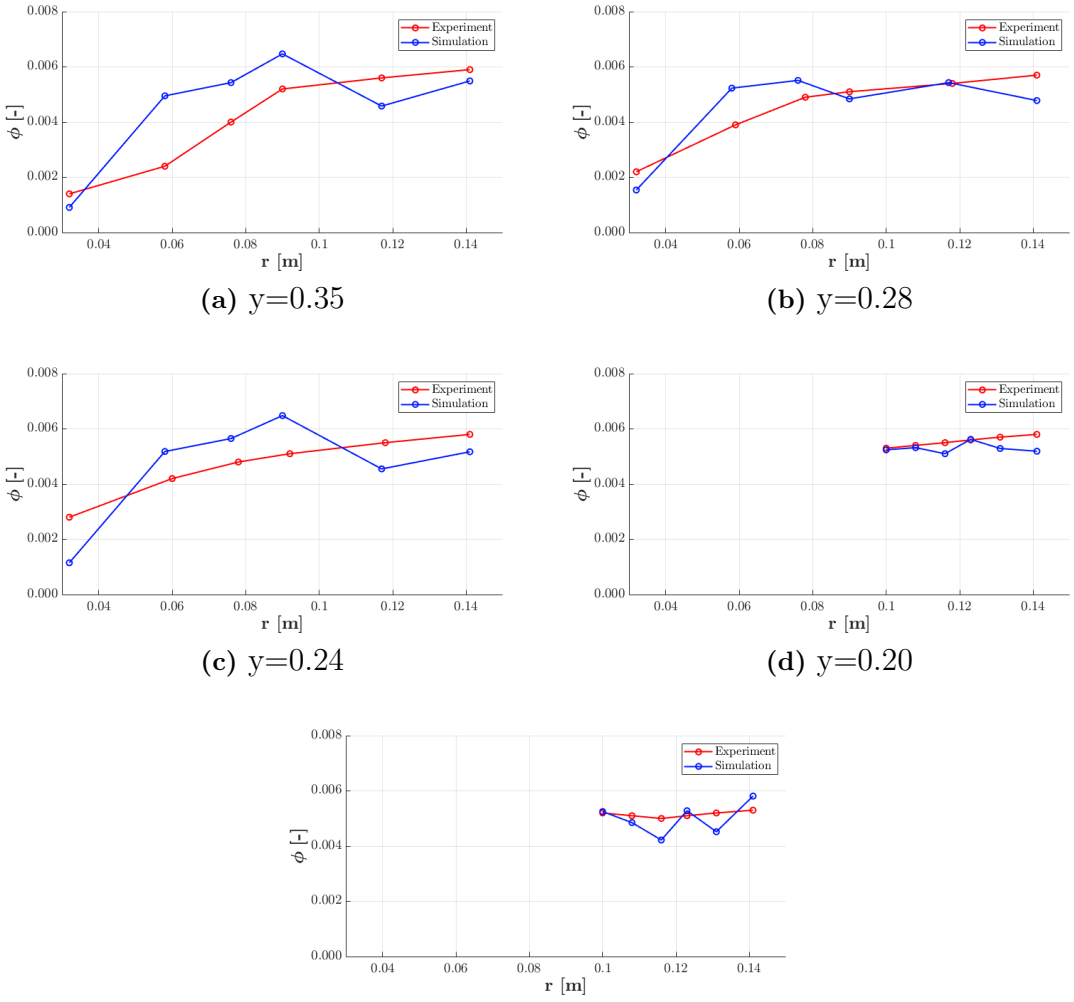
During the first few seconds after injection (Figures 4.18a–c), the particles rapidly disperse throughout the tank. By 5 s, the particles are more evenly distributed, and after this point, no major changes in the overall distribution are observed. As seen in Figures 4.18c and d, the particle distribution at 5 s is very similar to that at 30 s, indicating that the distribution in the tank stabilizes relatively quickly after injection



(a) t=1 s **(b)** t=3 s **(c)** t=5 s **(d)** t=30 s

Figure 4.18: Contour plots of solid volume fraction over time, LBM simulation.

To evaluate the particle distribution in the biomedical tank configuration of Case 3, solid volume fraction ϕ was sampled at a total of 30 probe points distributed throughout the tank. As shown in Figure 3.25, six probes were placed along the radial direction at each of five different heights: 0.14, 0.20, 0.24, 0.28, and 0.35 meters from the bottom of the tank.



(a) y=0.35 **(b)** y=0.28

(c) y=0.24 **(d)** y=0.20

(e) y=0.14

Figure 4.19: Solid volume fraction ϕ as a function of radial position r at five different heights in the tank ($y = 0.35$, 0.28, 0.24, 0.20, and 0.14 m) for the LBM simulation and corresponding experimental data from Shan et al. [1]. The impeller speed is 173 RPM.

Figure 4.19 presents the solid volume fraction ϕ as a function of radial position r for these five heights, comparing results from the LBM simulation with experimental data from Shan et al.[1]. The profiles show how the particle concentration varies radially at different vertical positions in the tank, and overall good agreement can be observed between the simulation and experimental data.

4.3.3 Solver time

Table 4.13 shows the solver time results for the different simulation approaches in case 3, including total CPU/GPU time and time normalized by simulated physical minutes. Only the LBM simulation was completed for this case, requiring a total of 5.87 hours of GPU time and 3.05 hours when normalized by simulated minutes.

Table 4.13: Solver time results for the different simulation approaches in case 3, including total CPU/GPU time and time normalized by simulated physical minutes.

	FVM Steady	**FVM Transient**	**LBM**
CPU/GPU time [h]	-	-	5.87
Normalized CPU/GPU time [h]	-	-	3.05

Although a steady-state FVM simulation was performed to resolve the flow field, it could not be combined with particle tracking and is therefore not included in the solver time comparison. The transient FVM simulation was initiated but could not be completed within the timeframe of the project. As a result, no meaningful comparison can be made between the FVM and LBM approaches in terms of particle-related simulation time for Case 3. Despite the complexity of the three-phase system, the LBM simulation was completed in a short time and yielded reliable results.

5

Discussion and Conclusion

This chapter presents a discussion of the simulation results and summarizes the main findings of the thesis. It also includes suggestions for future studies based on the challenges and observations encountered during the work.

5.1 Discussion

The results from case 1 indicate that the predicted mixing time from both the transient FVM and LBM simulations is in very good agreement with experimental data. This indicates that both methods can capture the overall mixing process with sufficient accuracy. However, LBM does not fully capture the length of the vortex, while the vortex predicted by FVM lacks any horizontal movement. The results for predicted mixing time indicate that the shape and position of the vortex do not severely impact the mixing process. Furthermore, the steady-state FVM approach clearly overestimated the mixing time, suggesting that transient flow characteristics play an important role in the mixing process. Both the FVM and LBM simulations also predicted the power number and pumping number with good accuracy.

Case 3 has the lowest aspect ratio between the tank and the impeller of all cases in this project, since the tank-to-impeller ratio is 2. This configuration has proven to be particularly advantageous for the LBM solver, as the low aspect ratio allows for a relatively coarse resolution, while still yielding highly accurate simulation results in a very short GPU time. Compared to case 1, where the tank-to-impeller ratio is 5, the normalized GPU time is about 9 times lower for case 3. Furthermore, the LBM solver shows promising potential for predicting particle distribution. However, since no corresponding FVM results are available for comparison, no definitive conclusions can be drawn regarding its relative performance in this aspect.

The results from case 2 show that using LBM in the selected software might not be not well suited for simulating large digester tanks, especially when long simulation times are required. The resolution study revealed that several key parameters did not fully converge before reaching memory limitations and unrealistic simulation times, which prevented further refinement. While impeller 1 showed reasonable agreement with the FVM results and generated physically realistic flow fields, impeller 2 did not. An unresolved flow field, in combination with a non-Newtonian fluid which

initialized at it's maximum viscosity, causes large regions with high viscosity in the tank since a certain velocity and shear rate is required for the fluid to act shear thinning. This limits the flow development, contributing to stagnant zones and inaccurate mixing time predictions.

The presence of high aspect ratios in the geometry introduces further difficulties, making the LBM solver less efficient. High aspect ratios often require local refinement, as demonstrated in the FVM simulations where a base size of 8% of the main mesh size was used in the MRF zone. This approach is more constrained when using the LBM solver, as the refinement strategy relies on halving the spatial resolution (dx) for each refinement level. Increasing the number of refinement levels significantly raises the computational cost, since the time step also is halved for each refinement level. Moreover, when refinement zones are applied, the D3Q27 lattice configuration must be used, which increases the computational demand by approximately 40%. For large geometries that already require substantial memory resources, conditions that are inherently suboptimal for GPU-based simulations, this additional computational burden quickly leads to unrealistic simulation times.

The LBM solver setup also includes limitations that further complicate simulations of large geometries. LBM inherently lacks support for steady-state solvers and the used LBM solver does not include simpler turbulence models like k-ε, meaning that all simulations are transient and relatively costly. Although this prioritizes accuracy, it also increases computational burden significantly, amplifying the existing issues. Previous studies suggest that such detailed flow resolution is not needed for realistic mixing predictions, which makes the LBM based solver less suitable under these conditions.

The difficulties encountered in case 2, such as high aspect ratios, the need for local refinement, high memory usage, and the inability to simplify the numerical setup, resulted in unrealistic simulation times and unreliable results. In contrast, the setup of case 1 was inherently more compatible with the strengths of LBM and GPU-based computation. The smaller geometry and lack of high aspect ratios reduce the memory intensity and the need for refinement, allowing the uniform lattice structure to be effectively parallelized on GPU hardware. The simpler domain allows the solver to maintain accuracy without excessive computational costs. These factors contribute to the good agreement observed between LBM, FVM, and experimental data in case 1, particularly in terms of predicted mixing time. Case 3 further demonstrates the strengths of the LBM solver, as its low tank-to-impeller aspect ratio enabled the use of a coarse lattice resolution while still achieving accurate particle distribution results within a very short simulation time. These results indicate that LBM is well suited for smaller tank systems with low memory demands and limited need for spatial refinement.

Comparison between CPU time for the different impeller speeds in case 1 suggests that the CPU time is strongly influenced by the number of cells in the mesh. The simulation with an impeller speed of 268 RPM was run on a mesh with approximately 20% more cells than the one used for the higher impeller speed. This higher cell

density resulted in an 80% longer CPU time, even though the time step size was larger for the lower impeller speed (due to the recommendation for sliding mesh). This clearly highlights one of the main disadvantages of FVM compared to LBM, where the mesh can have a significant impact on simulation cost. LBM uses a simple, uniform lattice structure instead of a complex, manually generated mesh, which avoids such sensitivity and contributes to a more consistent and predictable computational performance.

An important difference between FVM and LBM is the preprocessing effort required. In FVM, generating a high-quality mesh can be both time-consuming and demanding, and the CPU resources used during meshing are not included in the measured CPU times presented in the result. In contrast, LBM does not require mesh generation, which results in a faster and more straightforward setup, and contributes to the overall efficiency of the method.

5.2 Conclusion

This thesis has compared the performance of two numerical methods, FVM and LBM, in simulating mixing processes in fluid systems of different scales and complexities. The evaluation was conducted using STAR-CCM+ for FVM and M-Star for LBM, applied to two small-scale biomedical tanks and a large-scale anaerobic digester.

The results show that both methods can accurately predict the mixing time in the biomedical tank in case 1 and show good agreement with the experimental data. LBM was shown to be more computationally efficient than FVM for transient simulations. The simpler setup and uniform lattice structure in LBM also contributed to faster preprocessing and more consistent solver times. As expected, simulations run on the GPU were significantly more efficient for smaller fluid domains, likely due to better utilization of GPU memory and parallel processing. The comparison of the normalized GPU time between case 1 and case 3 shows that a tank with low aspect ratios allows for a coarser resolution, resulting in shorter simulation times without a significant trade-off between computational effort and accuracy.

However, the LBM solver showed clear limitations when applied to the digester tank. The simulation struggled with memory constraints, resolution convergence, and accurate prediction of impeller-related quantities, especially for impeller 2. The need for refinement zones, which in the LBM solver require use of the computationally expensive D3Q27 lattice, significantly increased the simulation time. Despite using a coarse grid setup, the simulations were still more time-consuming than the FVM simulations and failed to produce satisfactory results.

In conclusion, LBM is a suitable method for smaller geometries with limited need for spatial refinement, where fast setup and efficient computation are prioritized. For large-scale systems with high aspect ratios, requiring spatial refinement or extensive uniform resolution, FVM remains the more flexible and reliable choice.

5.3 Future Considerations

Future work should further investigate how surrounding geometry and grid resolution affect the accuracy of impeller simulations. In the study of case 2, only impeller 2 which is located near the draft tube, failed to produce realistic flow fields upon refinement, while impeller 1 did not show such issues. Since the draft tube is the main geometric difference between the two regions, its influence on the local flow conditions and mesh requirements requires deeper analysis.

Another aspect that could be interesting to investigate further is whether the vortex can be better resolved using the LBM solver. Since the vortex lacks depth in the current results, it could be useful to study the differences between the free surface model and the immiscible two fluid model in more detail. It could also be interesting to examine whether a higher resolution would improve the predicted vortex depth, if a GPU with more memory was available.

The LBM simulations for case 2 did not demonstrate time step size independence. Future research should investigate how the time step size affects the overall results, as this requires further analysis to ensure numerical stability and accuracy.

A possible solution to the challenges encountered in the digester tank simulations could be to use a uniform lattice with a higher resolution. In this study, this was not feasible due to memory limitations. Among all LBM configurations tested, the uniform lattice case produced the most developed flow field, with fewer high-viscosity regions and a distinct zone of higher velocity and lower viscosity around impeller 2, something not observed in the other setups. This suggests that a base resolution of 200 lattice points outside the refined zones was insufficient to resolve the entire flow field. Although a uniform lattice with more than 700 points could not be tested within the constraints of this thesis due to memory demands and excessive simulation times, this is a potential direction for future work. While memory usage is hardware-dependent, the high computational cost remains a major challenge. Further studies could assess whether a higher-resolution uniform lattice can produce a physically realistic flow field within reasonable simulation times.

Bibliography

[1] X. Shan, G. Yu, C. Yang, Z. S. Mao, and W. Zhang, "Numerical simulation of liquid-solid flow in an unbaffled stirred tank with a pitched-blade turbine downflow," *Industrial and Engineering Chemistry Research*, vol. 47, no. 9, pp. 2926–2940, 5 2008. [Online]. Available: https://pubs.acs.org/sharingguidelines

[2] L. Lindenbaum and J. Larsson, "CFD methods for liquid mixing with the presence of a vortex. A comparative study of simulational methods with experimental validation," 2023. [Online]. Available: http://hdl.handle.net/20.500.12380/306728

[3] M. A. Delavar and J. Wang, "Lattice Boltzmann method and its applications," *Handbook of HydroInformatics: Volume I: Classic Soft-Computing Techniques*, pp. 289–319, 1 2023.

[4] Siemens Digital Industries Software, "Simcenter STAR-CCM+, version 2021.1."

[5] "M-Star CFD documentation." [Online]. Available: https://docs.mstarcfd.com/index.html

[6] A. K. Pukkella and S. Subramanian, "An integrated CFD methodology for tracking fluid interfaces and solid distributions in a vortexing stirred tank," *Canadian Journal of Chemical Engineering*, vol. 101, no. 4, pp. 1854–1873, 4 2023.

[7] D. Dapelo, F. Alberini, and J. Bridgeman, "Euler-Lagrange CFD modelling of unconfined gas mixing in anaerobic digestion," *Water Research*, vol. 85, pp. 497–511, 11 2015.

[8] C. Sadino-Riquelme, R. E. Hayes, D. Jeison, and A. Donoso-Bravo, "Computational fluid dynamic (CFD) modelling in anaerobic digestion: General application and recent advances," *Critical Reviews in Environmental Science and Technology*, vol. 48, no. 1, pp. 39–76, 1 2018. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/10643389.2018.1440853

[9] K. J. Craig, M. N. Nieuwoudt, and L. J. Niemand, "CFD simulation of anaerobic digester with variable sewage sludge rheology," *Water Research*, vol. 47,

no. 13, pp. 4485–4497, 9 2013.

[10] F. J. Hurtado, A. S. Kaiser, and B. Zamora, "Fluid dynamic analysis of a continuous stirred tank reactor for technical optimization of wastewater digestion," *Water Research*, vol. 71, pp. 282–293, 3 2015.

[11] E. Wicklein, D. J. Batstone, J. Ducoste, J. Laurent, A. Griborio, J. Wicks, S. Saunders, R. Samstag, O. Potier, and I. Nopens, "Good modelling practice in applying computational fluid dynamics for WWTP modelling," *Water Science and Technology*, vol. 73, no. 5, pp. 969–982, 3 2016. [Online]. Available: https://www.researchgate.net/publication/295241928_Good_modelling_practice_in_applying_computational_fluid_dynamics_for_WWTP_modelling

[12] B. Wu, "CFD investigation of turbulence models for mechanical agitation of non-Newtonian fluids in anaerobic digesters," 2010.

[13] A. M. Karpinska and J. Bridgeman, "CFD-aided modelling of activated sludge systems – A critical review," *Water Research*, vol. 88, pp. 861–879, 1 2016.

[14] R. A. Hoffmann, M. L. Garcia, M. Veskivar, K. Karim, M. H. Al-Dahhan, and L. T. Angenent, "Effect of shear on performance and microbial ecology of continuously stirred anaerobic digesters treating animal manure," *Biotechnology and Bioengineering*, vol. 100, no. 1, pp. 38–48, 5 2008. [Online]. Available: https://www.researchgate.net/publication/5824222_Effect_of_shear_on_performance_and_microbial_ecology_of_continuously_stirred_anaerobic_digesters_treating_animal_manure

[15] F. Shen, L. Tian, H. Yuan, Y. Pang, S. Chen, D. Zou, B. Zhu, Y. Liu, and X. Li, "Improving the mixing performances of rice straw anaerobic digestion for higher biogas production by computational fluid dynamics (CFD) simulation," *Applied Biochemistry and Biotechnology*, vol. 171, no. 3, pp. 626–642, 10 2013. [Online]. Available: https://www.researchgate.net/publication/250924590_Improving_the_Mixing_Performances_of_Rice_Straw_Anaerobic_Digestion_for_Higher_Biogas_Production_by_Computational_Fluid_Dynamics_CFD_Simulation

[16] R. Sindall, J. Bridgeman, and C. Carliell-Marquet, "Velocity gradient as a tool to characterise the link between mixing and biogas production in anaerobic waste digesters," *Water Science and Technology*, vol. 67, no. 12, pp. 2800–2806, 2013. [Online]. Available: https://www.researchgate.net/publication/240308757_Velocity_gradient_as_a_tool_to_characterise_the_link_between_mixing_and_biogas_production_in_anaerobic_waste_digesters

[17] "Fluent 6.3 Getting Started Guide: September 2006 | PDF | Search Engine Indexing | Command Line Interface." [Online]. Available: https://www.scribd.com/document/43512708/flgsqw

[18] A. W. Nienow, M. F. Edwards, and N. Harnby, "Introduction to mixing prob-

lems," pp. 1–24, 1 1992.

[19] B. Sajjadi, A. Aziz, A. Raman, and R. Parthasarathy, "Fluid dynamic analysis of non-Newtonian flow behavior of municipal sludge simulant in anaerobic digesters using submerged, recirculating jets," 2016. [Online]. Available: http://dx.doi.org/10.1016/j.cej.2016.03.069

[20] T. Neuner, M. Meister, M. Pillei, T. Senfter, S. Draxl-Weiskopf, C. Ebner, J. Winkler, and W. Rauch, "Impact of Design and Mixing Strategies on Biogas Production in Anaerobic Digesters," *Water (Switzerland)*, vol. 16, no. 15, 8 2024.

[21] D. Dapelo and J. Bridgeman, "Euler-Lagrange Computational Fluid Dynamics simulation of a full-scale unconfined anaerobic digester for wastewater sludge treatment," *Advances in Engineering Software*, vol. 117, pp. 153–169, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.advengsoft.2017.08.009

[22] "Handbook of industrial mixing - Search results - EBSCO Discovery Service." [Online]. Available: https://research.ebsco.com/c/lu54te/search/results?limiters=&q=Handbook+of+industrial+mixing

[23] H. Singh, D. F. Fletcher, and J. J. Nijdam, "An assessment of different turbulence models for predicting flow in a baffled tank stirred with a Rushton turbine," 2011. [Online]. Available: www.elsevier.com/locate/ces

[24] A. Tamburini, G. Gagliano, G. Micale, A. Brucato, F. Scargiali, and M. Ciofalo, "Direct numerical simulations of creeping to early turbulent flow in unbaffled and baffled stirred tanks," *Chemical Engineering Science*, vol. 192, pp. 161–175, 12 2018.

[25] L. Li and B. Xu, "CFD simulation of local and global mixing time in an agitated tank," *Chinese Journal of Mechanical Engineering (English Edition)*, vol. 30, no. 1, pp. 118–126, 1 2017. [Online]. Available: https://link.springer.com/articles/10.3901/CJME.2016.1107.129https://link.springer.com/article/10.3901/CJME.2016.1107.129

[26] B. Andersson, R. Andersson, L. Hakansson, M. Mortensen, R. Sudiyo, and B. Van Wachem, *Computational fluid dynamics for engineers*. Cambridge, England: Cambridge University Press, 12 2011.

[27] S. Chakraverty and others, *Advanced Numerical and Semi-Analytical Methods for Differential Equations*. John Wiley & Sons, Incorporated, 2019. [Online]. Available: https://ebookcentral.proquest.com/lib/chalmers/detail.action?docID=5741224

[28] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, "The Lattice Boltzmann Method," 2017. [Online]. Available: http://link.springer.com/10.1007/978-3-319-44649-3

[29] Z. Guo and C. Shu, *Lattice Boltzmann Method and Its Application in Engineering*. World Scientific Publishing Company, 2013.

[30] F. Concli and C. Gorla, "Analysis of the power losses in geared transmissions - measurements and CFD calculations based on open source codes," *International Gear Conference 2014: 26th–28th August 2014, Lyon*, pp. 1131–1140, 1 2014.

[31] J. Tu, G.-H. Yeoh, and C. Liu, "Some Advanced Topics in CFD," in *Computational Fluid Dynamics*. Elsevier, 2018, pp. 369–417.

[32] J.-D. Mueller, *Essentials of computational fluid dynamics*. London, England: CRC Press, 12 2020.

[33] P. Tsoutsanis and X. Nogueira, "Arbitrary-order unstructured finite-volume methods for implicit large eddy simulation of turbulent flows with adaptive dissipation/dispersion adjustment (ADDA)," *Journal of Computational Physics*, vol. 523, p. 113653, 2 2025.

[34] "Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications," 2008. [Online]. Available: http://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/130/7/078001/5491455/078001_1.pdf

[35] A. Soroushian, "A DIFFERENT LOOK AT THE RICHARDSON EXTRAPOLATION LEADING TO A NEW PROPOSITION," no. 21, 1953.

[36] M-STAR CFD, "Theory and Implementation," 2024. [Online]. Available: https://docs.mstarcfd.com/17_Theory/theory-and-implementation.html

[37] T. Wang, H. Wang, and G. Tang, "Benchmarking the Star-CCM+ compressible flow solver by simulating typical compressible flow problems: A case study and comparison," *Communications in Computer and Information Science*, vol. 645, pp. 379–391, 2016. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-10-2669-0_41

[38] A. Syberfeldt and T. Ekblom, "A Comparative Evaluation of the GPU vs The CPU for Parallelization of Evolutionary Algorithms Through Multiple Independent Runs," *International Journal of Computer Science and Information Technology*, vol. 9, no. 3, pp. 01–14, 6 2017.

[39] Siemens, "Specifying Prism Layer Size on the Interface," Online, 2024. [Online]. Available: https://docs.sw.siemens.com/documentation/external/PL20191230144651718/en-US/userManual/userGuide/html/index.html#page/STARCCMP%5C%2FGUID-ADFCC925-E117-4630-8D86-13C171962405.html%5C%23.

A

Results from case 1, impeller speed 268 RPM

A.0.1 Vortex

Figure A.1 shows the vortex shape in the FVM and LBM simulations at 268 RPM.

(a) Transient FVM simulation **(b)** LBM simulation

Figure A.1: Vortex shape captured in the FVM transient and LBM simulations at an impeller speed of 268 RPM.

Figure 4.3 shows the average vortex from the FVM simulation at 268 RPM.

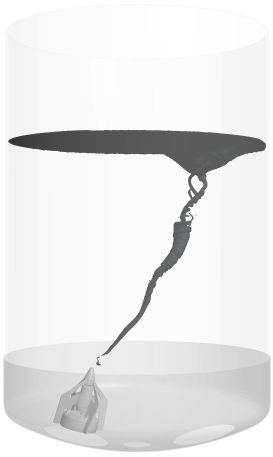
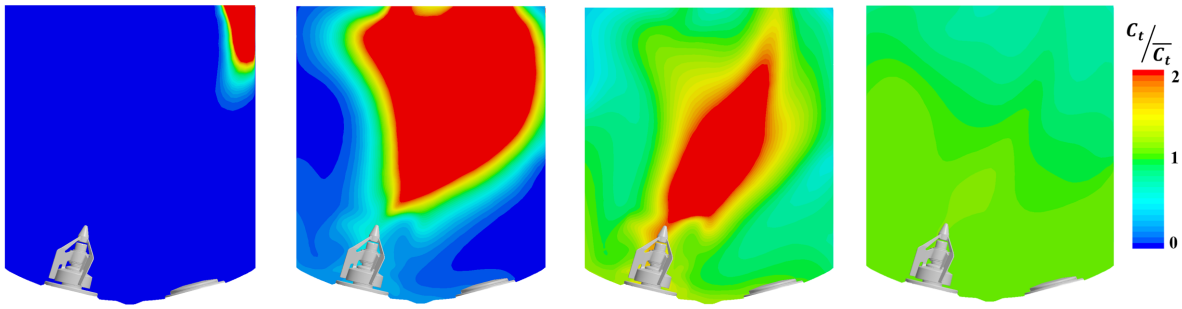


Figure A.2: Time-averaged vortex surface from the transient FVM simulation at 268 RPM.

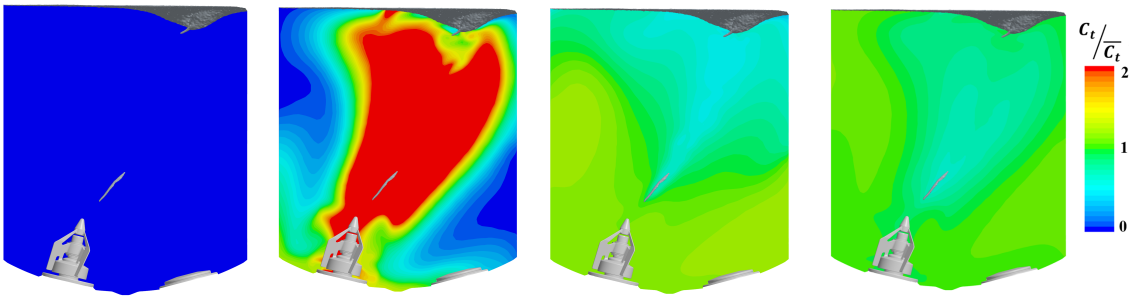
A.0.2 Transport of the passive scalar for impeller speed of 268 RPM.

The mixing process is illustrated in the contour plots of normalized concentration shown in Figures A.3, A.4, and A.5, which depict how the tracer spreads throughout the domain over time.



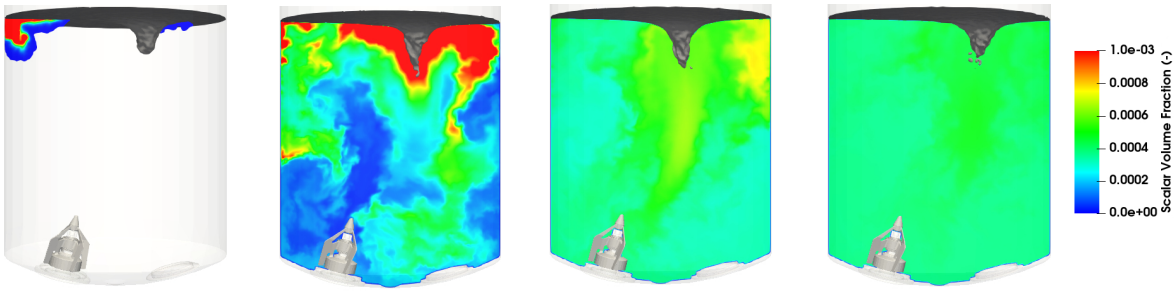
(a) t=2 s (b) t=10 s (c) t=18 s (d) t=25 s

Figure A.3: Contour plots of normalized concentration over time, FVM steady simulation.



(a) t=2 s (b) t=10 s (c) t=18 s (d) t=25 s

Figure A.4: Contour plots of normalized concentration over time, FVM transient simulation.

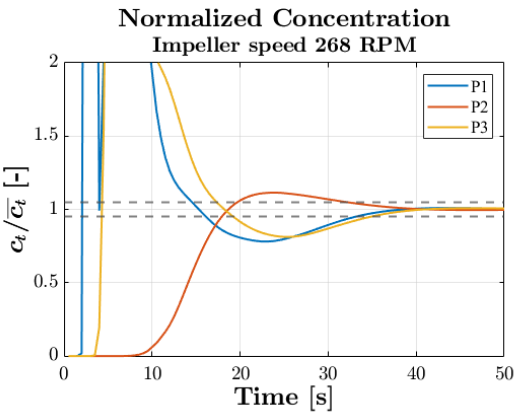


(a) t=2 s (b) t=10 s (c) t=18 s (d) t=25 s

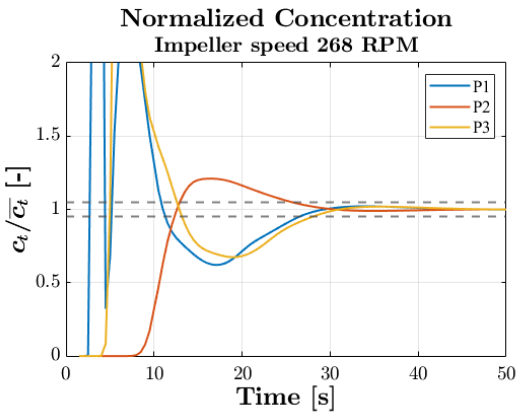
Figure A.5: Contour plots of scalar volume fraction over time, LBM simulation.

A.0.3 Normalized concentration along probes for impeller speed of 268 RPM.

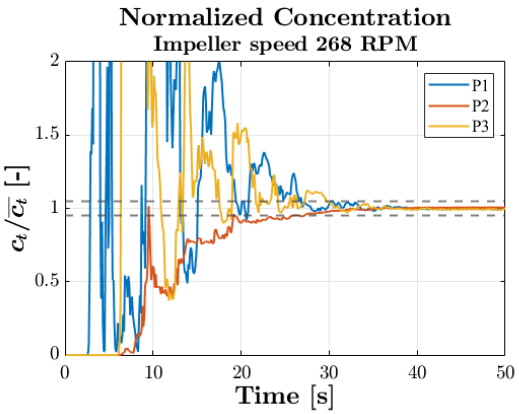
Figure A.6 shows the time evolution of the normalized concentration at three probe locations for an impeller speed of 268 RPM.



(a) Normalized tracer concentration at three probe points from the steady-state FVM simulation.



(b) Normalized tracer concentration from the transient FVM simulation.



(c) Normalized tracer concentration from the LBM simulation.

Figure A.6: Time evolution of normalized tracer concentration at three probe locations for the different simulation approaches at 268 RPM. Mixing is considered complete when the normalized concentration at all three probes reaches and remains within the interval 0.95–1.05. This interval is indicated by grey dashed horizontal lines in all plots.

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY