Dipl. Ing. Christian Witz BSc

GPU-powered Simulation of Industrial-Scale Aerated Stirred Tank Bioreactors

DOCTORAL THESIS

to achieve the university degree of

"Doktor der technischen Wissenschaften"

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Johannes G. Khinast

Institute for Process and Particle Engineering

Graz, December 2016

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

_____ _____

Date Signature

To my family

Sandra & Gustav

Besser verwirrt sein als gar nicht denken.

Tocotronic

CONTENTS

ABSTRACT

Aerated stirred tank reactors are widely used in many industry sectors. Despite this fact, the optimization of the operational costs and the production efficiency is still mostly based on empirical knowledge. Computer simulation might empower the engineers to better understand the processes inside the reactors. With the calculation power of modern general purpose graphic processing units (GPGPUs) and the Compute Unified Device Architecture (CUDA) programming language the Computational Fluid Dynamics (CFD) simulation of large scale reactors becomes possible for the first time. The present work enables to fully utilize this new developments of computational sciences for solving the aforementioned optimization problem in process engineering and industry. The algorithms and model concepts applied were carefully selected for providing high parallelizability and memory efficiency.

For the fluid flow field calculation the lattice Boltzmann algorithm was used as it is highly localized and hence achieving minimal memory access times on GPUs. To simulate the gas bubble movement with the Lagrangian approach the Newton's equations of motion were solved. The liquid and gas phases were two-way coupled. The breakup and the coalescence of the bubbles are modelled applying a stochastic approach. The probability of bubble breakup or coalescence events depends on the rate of approaching turbulent eddies and the comparison of the film breakage time with the contact time. The simulation code was validated with literature data and gas holdup measurements using a conductivity sensor in a 150 l custom-built acrylic reactor.

Stirring and aeration of the bioreactor supports converting the dissolved substrates such as sugar and oxygen to the desired product and to side products such as carbon dioxide by microorganisms. To model the mass transfer of substrates and products in the bioreactor a species transport solver algorithm based on the lattice Boltzmann method was developed and implemented. The oxygen transfer from gas bubbles into the liquid phase and the uptake of carbon dioxide by the gas bubbles are modelled applying the penetration theory of Higbie. A simple biological model was implemented to demonstrate the capability of the code to simulate the substrate uptake and product formation as the effect of the microbial metabolism.

KURZFASSUNG

Begaste und gerührte Bioreaktoren sind in vielen Industriesektoren verbreitet. Dennoch wird die Optimierung der Betriebskosten und der Produktionseffizienz häufig auf Basis von empirischen Erfahrungswerten durchgeführt. Computersimulationen können den Ingenieuren helfen die Prozesse im Inneren der Reaktoren besser zu verstehen. Mit der Rechenleistung moderner Grafikkarten und der Programmiersprache CUDA kann die Simulation von Reaktoren im Industriemaßstab erstmals durchgeführt werden. In dieser Arbeit werden diese computerwissenschaftlichen Neuentwicklungen verwendet um das zuvor genannte Optimierungsproblem bestmöglich zu lösen. Die verwendeten Algorithmen wurden hinsichtlich ihrer Parallelisierbarkeit und ihrer Effizienz der Speichernutzung ausgewählt.

Zur Berechnung des Strömungsfelds der Flüssigkeit wird die Lattice Boltzmann Methode verwendet, da diese nur lokale Variablen zur Berechnung des Geschwindigkeitsfelds nutzt und daher minimale Speicherzugriffszeiten auf Grafikkarten erzielt. Die Gasblasenbewegung wird mit dem Lagrangen Ansatz durch Aufsummierung der Newtonschen Bewegungsgleichungen berechnet. Die Gas- und die Flüssigphasen sind wechselseitig gekoppelt. Die Dispersion und die Koaleszenz der Blasen werden mit einem statistischen Ansatz approximiert. Die Rate der auftreffenden Turbulenzwirbel bestimmt die Häufigkeit der Teilungen der Blasen. Ein Vergleich der Kontaktzeit bei Blasenkollisionen und der Zeit, die der Flüssigkeitsfilm zwischen den kollidierenden Blasen zum Reißen benötigt, ergibt die Häufigkeit der Fälle von Koaleszenz. Der Simulationscode wurde über einen Vergleich der Dispersphasenanteile validiert. Die experimentellen Daten hierfür wurden mittels eines Leitfähigkeitssensors in einem 150 l Plexiglasreaktor gewonnen.

Der Zweck eines Bioreaktors ist es die Mikroorganismen dabei zu unterstützen das eingesetzte Substrat wie beispielsweise Zucker und den gelösten Sauerstoff in das gewünschte Produkt und in Nebenprodukte wie etwa Kohlendioxid umzuwandeln. Um diesen Prozess zu modellieren wurde ein Transportgleichungslöser entwickelt und implementiert. Der Stoffübergang des Sauerstoffs in die flüssige Phase sowie die Aufnahme des gelösten Kohlendioxids durch die Luftblasen wurde durch das Penetrationsmodell von Higbie berechnet. Ein einfaches biologisches Modell wurde implementiert um zu demonstrieren, dass die Stoffwechselvorgänge von Mikroorganismen mit diesem Code simuliert werden können.

ACKNOWLEDGEMENT

First and foremost, I would like to thank my advisor Prof. Johannes Khinast for the opportunity to work on this very interesting topic, for his continuous personal and financial support during my PhD and for motivating me when programming bugs were testing my patience. I would also like to thank him for the opportunity to attend and speak at international conferences. Besides my advisor I would like to thank Prof. Siebenhofer for being the second assessor of my thesis.

This project wouldn't have been possible without the continuous support of our business partner Sandoz represented by Timo Hardiman, Dirk Behrens, Sören Bernauer and Hubert Kürnsteiner. I especially thank Timo Hardiman for providing valuable comments on the scientific papers and on this thesis.

My sincere thanks also goes to Stefan Radl for answering numerous questions, for providing validation cases to test the code and for organizing simulation group meetings where the discussions about the research topics provided state-of-the-art insights in other areas in the field of simulation science. Also I thank Daniel Treffer, who has helped me many times with the setup of the acrylic glass reactor and the conductivity sensors. Thanks to Hans Grubbauer who continuously supported my students and me in the lab. I also want to express my gratitude to Tawan Tantikul, who led the project when I started at the IPPT and helped me a lot acquiring the knowledge necessary to complete this thesis.

I thank my students Annika Grafschafter for automating the reactor operation, Benedict Benque for including and testing non-Newtonian fluid algorithms and Philipp Eibl for programming modules for the shear rate, the power number, advanced stirrers, heat exchangers and sensors. Their excellent work has been an important contribution to my PhD.

I want to specially thank all the present and past members of the institute: Michaela Cibulka, Simone Eder, Andreas Eitzlmayr, Piet Feenstra, Thomas Forgber, Bianca Grabner, Heidi Gruber-Wölfler, Johannes Gursch, Theresa Hörmann, Silvia Houben, Stefan Karner, Georg Lichtenegger, Peter Loidolt, Josip Matic, Bhageshvar Mohan, Federico Municchi, Peter Neugebauer, Jakob Redlinger-Pohn, Adela Roller, Stefan Scheer, Mingqiu Wu and Sarah Zellnitz for all the support I received and for all the fun we had in the office, especially at coffee and lunch breaks.

For giving me invaluable advice on choosing my advisor and the PhD program, for always reminding me how grateful one can be to be able to conduct scientific research as well as for numerous discussions on topics both inside and outside my PhD thesis, I would like to express my gratitude to Prof. Reinhart Kögerler.

Meinem Onkel Ulrich Schneebauer danke ich für die vielen Gespräche und Essenseinladungen.

Mein besonderer Dank gilt meinen Eltern, die mir stets mit Rat und Tat zur Seite standen, mir das Studium ermöglichten und mir mit sehr viel Ausdauer das Rüstzeug mitgaben, nun auch selbst ein Elternteil zu sein.

Meinem Bruder Gerhard danke ich für die Unterstützung während des Doktorats, für die vielen Gespräche und Ratschläge.

Von ganzem Herzen bedanke ich mich bei meiner Frau Sandra die mir in allen Phasen der Dissertation zur Seite gestanden ist und mich immer wieder motiviert und bestärkt hat. Ohne sie wäre diese Arbeit nicht möglich gewesen, deshalb ist diese Dissertation auch unserem Sohn Gustav und ihr gewidmet.

Christian Witz

Graz, December 2016

1
Introduction to Multiphase Reactor Modelling

1.1 Overview

Multiphase reactors are widely used in the chemical industry. Common types are stirred tank reactors to promote the fluid phase homogenization as well as to increase the interface between the fluid and the gaseous phase by breaking up the gas bubbles. The increased surface as well as the higher relative velocity between the phases is beneficial for the mass transfer of components of the gaseous phase into the liquid. In the case of the bioreactors, a large amount of dissolved oxygen is needed to enable the aerobic cultivation of microorganisms.

The reactors are used in many different sizes and operating conditions. These conditions as well as the geometry of the reactor itself, the baffles and the stirrer type are subject to optimization. The goal of the optimization besides providing ideal conditions for the microorganisms is to reduce the energy required to power the stirrer motor and to compress the air for the sparger. Today this optimization is mainly guided by empirical knowledge. Computer simulation in the form of Computational Fluid Dynamics (CFD) is an opportunity to provide a more efficient and robust way for optimization and scaling procedures based on scientific principles.

The first step to enable modeling the complex dynamic processes inside such reactors is to simulate the turbulent liquid phase flow field. The partial differential equations that describe the movement of fluids are called the Navier Stokes equations. Unfortunately, the direct analytical solution of these equations is only possible for a very limited number of cases with special boundary conditions (Wang, 1991).

To simulate general cases of fluid movement – as for stirred tank reactors – the equations have to be discretized. One way which has become popular recently is the lattice Boltzmann method (Chen and Doolen, 1998a).

The calculation domain – in case of the reactor a cuboid enclosing the cylindrical walls – is divided regularly into cubes by equidistant calculation points. Hence, a complex grid generation procedure is unnecessary. Each of those points is equipped with a set of density distribution vectors pointing in different spatial directions. The value assigned to the vectors corresponds to the probability that the fluid mass in the cube moves into the spatial direction the vector is pointing at in the current time step. This value is depending on external forces and the macroscopic fluid velocity which is embodied in the so-called equilibrium function. The density distribution vector is relaxed towards the equilibrium value. The relaxation factor, which is calculated using the viscosity, determines the strength of this relaxation. After this calculation, which is known as the collision step, the distribution vectors are copied to the neighboring nodes, retaining their direction. After this step, called the streaming step, the density and the macroscopic fluid velocity are calculated by summing the values of the vectors.

The movement of the bubbles is determined by the sum of all forces like drag, lift, added mass force as well as fluid stresses and gravity. The drag, lift and the added mass force are coupled back to the liquid phase to achieve a two-way coupling. This is important for bioreactors with high volume fractions of the gaseous phase (gas holdup) since this disperse phase can strongly influence the velocity field of the continuous liquid phase. To estimate the interphase area between the phases correctly the breakup and coalescence of the bubbles has to be modelled.

When all the effects on the multiphase flow field described above are represented in a CFD model the most interesting process can be simulated: the species transport. The distribution of nutrients like glucose and the dissolved oxygen as well as undesired side-product accumulation such as carbon dioxide and, most interestingly, the effects on the formation rate of the desired product inside the reactor can be described using a (biological) reaction model and calculating the species transport.

However, the modelling of the species transport is not straightforward. The classical approach includes the discretization of the transport equation which is not easy to parallelize. The use of Lagrangian particles is another possibility to calculate the concentration field which is better parallelizable. A small amount of species mass is assigned to each of the particles which are considered massless (i.e. not affected by gravity, added mass, drag, etc.) and are transported by the fluid flow field. Unfortunately, the memory requirement for storing the particles positions and velocities is very high and is directly related to the precision of the concentration field resolution.

A method which would combine efficient parallelization and moderate memory consumption is the lattice Boltzmann method. Unfortunately the inherent characteristics of the method might lead to negative concentration values and mass conservation problems (Karimi and Nakshatrala, 2016). Additionally, the vector based distribution of the species mass can lead to an artificial inertia which is unphysical. These well-known problems are discussed in the present thesis and a possible solution is introduced.

1.2 This Thesis

The goal of this thesis was to provide a code which is able to simulate the transient processes inside industrial-size bioreactors. All models are based on elementary physical principles. Empirical correlations were avoided. One of the novelties of this work is that modern graphic cards are used as a hardware platform to reduce the time for completing the simulation significantly and hence render it suitable for using it in the engineering process of novel reactor designs. All algorithms were particularly selected to preserve this advantage and to fully utilize the graphic processors (GPUs). The Euler-Lagrange approach was applied to model the liquid flow field and its interaction with the gas phase. The advantage of this approach is knowledge of the exact locations and diameters of the gas bubbles, which is beneficial for the modelling of the oxygen transfer, the correct local distribution of the power input, and therefore the resulting flow regime depending on stirring and aeration conditions.

The second chapter explains the used lattice Boltzmann algorithm for the fluid field calculation as well as the boundary conditions. The bubble movement calculation using the Lagrangian particle tracking approach is presented including the models for bubble breakup and coalescence. The simulation time is compared for the simulation on one and two GPUs. Validation is illustrated based on literature data.

The third chapter discusses the application of the simulation model to pilot scale reactors. To gather further validation data a conductivity sensor was used in a custom-built 150l acrylic reactor. The rotational stirrer speeds and the gas flow rates of several flow regimes were studied. The disperse gas phase volume fraction was measured at

different radial and vertical distances and compared to the predicted values of the simulation. Good agreement between the measured and the predicted data was achieved.

The fourth chapter deals with the development and testing of a species transport solver based on the lattice Boltzmann method. The resulting solver is easily parallelizable and runs efficiently on GPUs. After the validation with a Gaussian peak transport the solver was used for the transient transport of nutrients in the aerated and stirred bioreactor, for the comparison of single and split feed and for the dissolution and transport of dissolved oxygen. To demonstrate the ability to simulate the metabolism of microorganism, a simple biological model was implemented.

Conclusion and outlook are discussed in chapter five. The key points of the research are summarized and a prospect on further research is given.

1.3 References

Chen S, Doolen GD. 1998. Lattice Boltzmann Method for Fluid Flows. *Annu. Rev. Fluid Mech.* **30**:329–364.

Karimi S, Nakshatrala KB. 2016. Do Current Lattice Boltzmann Methods for Diffusion and Advection-Diffusion Equations Respect Maximum Principle and the Non-Negative Constraint? *Commun. Comput. Phys.* **20**:374–404.

Wang CY. 1991. Exact Solutions Of The Steady-State Navier-Stokes Equations. *Annu. Rev. Fluid Mech.* **23**:159–177.

White FM. 1974. Viscous fluid flow 3rd ed. *New York:* McGraw-Hill

2

Bioreactor Simulation with CUDA[1]

An algorithm to simulate the fluid flow field and the bubble movement inside a large aerated bioreactor with a Rushton turbine was developed. A two-way coupling between the lattice Boltzmann method and the Lagrangian particle tracking was applied to model the dynamics of the liquid and the gas phase. The boundaries were included by the modified bounce back approach. Bubble breakup and coalescence are modelled using the local statistics around the bubble location. Good agreement between the predicted averaged fluid and gas velocities was achieved. As the chosen calculation algorithms can be efficiently parallelized a large speed-up was achieved by using the Compute Unified Device Architecture (CUDA) technology of Nvidia to execute the code on a graphic processing unit (GPU). Additional reduction of the simulation time was achieved by distributing the simulation to two GPUs which resulted in a reduction of simulation time by the factor of 30. This modelling technique leads to a simulation tool suitable for the use in bioreactor design.

[1] This chapter is based on Witz C, Khinast JG. 2015. Bioreactor simulation with CUDA. In: Schindler, F-P, Kraume, M; Fortschritt-Berichte VDI Reihe 3. Düsseldorf: VDI-Verlag, pp. 91–105.

2.1 Introduction

A critical challenge in the biopharmaceutical production process is the design, optimization and scale-up of bioreactors. In the reactor a large number of simultaneous processes occur, such as the flow of the liquid and the gas phase, macro-, meso and micro-mixing and dissolution of oxygen, as well as the metabolism of microorganisms or cells. To understand and optimize the performance of industrial reactors empirical knowledge can be combined with a multiphase and multi-scale simulation of the processes in the reactor. Simulation results may then be used for the development and design of the reactor system. However, despite the recent improvement of computational capabilities, the simulation of only a few seconds of real-time operation of an industrial-scale reactor usually takes months. The development of more efficient methods is thus critical for progress in this field.

Eggels (Eggels, 1996) was one of the first to use the Lattice-Boltzmann Method (LBM) with a Smagorinsky subgrid turbulence model to simulate turbulent flows in a stirred reactor. The method is based on the lattice gas automata and uses a regular grid with evenly distributed nodes. In contrast to our work (19 distribution functions), no resting distribution function is used in Eggels' LBM scheme (18 distribution functions). The impeller was modelled as forces acting on the fluid field. Derksen and Van den Akker (Derksen and van den Akker, 1999) used the adaptive force field technique to model the static and dynamic boundaries. Euler – Lagrange simulations of stirred reactors were done by Arlov et al. (Arlov et al., 2008), who used the filtered incompressible Navier–Stokes equations and Lagrangian particle tracking with monodisperse bubbles. Sungkorn et al. (Sungkorn et al., 2011; Sungkorn et al., 2012a; Sungkorn et al., 2012b)

applied LBM of Eggels to model the fluid field and included a breakup and coalescence model in the Lagrangian particle tracking technique. Han et al. (Han et al., 2007) reported the use of the discrete particle method to model the dispersed phase.

The Compute Unified Device Architecture (CUDA) technology of Nvidia has made the computational power of graphic processing units (GPUs) available for scientific calculations. To take full advantage of the processor's parallelism, localized calculation algorithms are needed. The studies of Obrecht et al. (Obrecht et al., 2011; Obrecht et al., 2013b) and Tölke (Tölke, 2008) have shown the large potential to speed up simulations when the lattice Boltzmann method is used on graphic processors. Thus, the goal of our current study was to use graphic-card technology to enable two-phase simulations of large-scale systems in the reactor engineering process.

In our work, LBM is used for the modelling of the fluid flow field using 19 statistical distribution functions and the Bhatnagar-Gross-Krook (Qian et al., 1992) approximation. To model the turbulence the Smagroinsky subgrid model (Smagorinsky, 1963) is included. The static and moving boundaries are modelled with the modified bounce back algorithm. The calculation of the bubble movement is done by solving Newton's equations of motion. The sum of the forces acting on each bubble, i.e., drag, buoyancy, lift force, history force, added mass effect and gravity determines the acceleration of the bubbles. The acceleration and the time step length give the velocity and the position change at the end of the time step. For coalescence and breakup stochastic models are used. The phases are coupled with a two way approach.

To simulate large reactors, the code has multi-GPU functionality. Hence it can distribute the workload of the simulation on several graphic processors. Validation is done with literature data.

In the next section the numerical algorithms mentioned above will be described and discussed in detail. The subsequent chapter will show some of the results of the validation test cases, as well as the full reactor simulation. Finally, a summary will be given in the conclusion section.

2.2 Numerical modelling

2.2.1 Fluid flow field calculation

One of the possibilities to calculate fluid flow fields is the use of the lattice Boltzmann method, which was developed from lattice gas automata (Sukop and Thorne, 2006). It uses a regular grid with evenly distributed nodes. Around each node (a) statistical distribution functions (f_a) are arranged in several spatial directions (e_a), i.e., 19 in this study. Each of them has a specific weighting factor (w_a).

Those functions together with an equilibrium function (f_a^{eq}) recover the macroscopic Navier Stokes equations. The calculation consists of two steps. In the first step, the collision step, the difference between the current and the aforementioned equilibrium distribution function is divided by the relaxation factor τ (Formula (1)). This is known as BGK (Bhatnagar-Gross-Krook) approximation (Qian et al., 1992). These updated distribution functions are then streamed to the next node into the direction they are pointing to. Hence the name “streaming step”.

$$f_a\left(x + e_a \Delta t, t + \Delta t\right) = f_a\left(x,t\right) - \frac{f_a\left(x,t\right) - f_a^{eq}\left(x,t\right)}{\tau} - 3 w_a F \cdot e_a \tag{1}$$

$$f_a^{eq}\left(x\right) = w_a \rho\left(x\right)\left[1 + 3\frac{e_a \cdot u}{c^2} + \frac{9}{2}\frac{\left(e_a \cdot u\right)^2}{c^4} - \frac{3}{2}\frac{u^2}{c^2}\right] \tag{2}$$

$$\rho = \sum f_a \tag{3}$$

$$\rho u = \sum f_a e_a \tag{4}$$

The sum of the distribution functions represents the density (ρ) (3); the direction-weighted sum gives the product of the density and the macroscopic velocity (u) (4). The major advantage of LBM in combination with CUDA is that every calculation and read access is done on the node. Only in the streaming step variables are written to neighbor nodes. Therefore, the method can be parallelized in a straightforward manner.

Several strategies to model the boundaries have been reviewed and some of them tested in the present work. In the immersed boundary method the boundary is represented with evenly distributed points (Goldstein et al., 1993). The fluid velocity is interpolated to the location of those points. A force is then calculated, which, if applied to the fluid field, requires the flow velocity to be zero at the boundary points. In the explicit form, this method led to leak flows in the present work (data not shown) due to the quickly changing and fluctuating flow field, which would cause mass loss when the flow field is used in a species transport solver calculation. The implicit form of the immersed boundary method (Wu and Shu, 2009) might circumvent this drawback. However, as a linear equation system has to be solved the method needs far more computational time than the modified bounce back method (MBB) (Ladd, 1994) and the extrapolation method (EM) (Guo et al., 2002b) while producing comparable results (Stobiac et al., 2013).

The above facts are the reason why the MBB method was chosen in this study. It is a half-way bounce back algorithm which modifies the statistical distribution functions. If

the boundary is static, the distribution functions are reflected. For moving boundaries like stirrer blades, which have a solid part velocity u_b, the following formula is used:

$$f_{a'}(x,t+\Delta t) = f_a(x,t) - 6 w_a \rho u_b e_a \tag{5}$$

Here, $f_{a'}$ is the distribution function after the bounce back and points in the opposite direction of f_a. The solid boundary is considered half way between two fluid nodes. As the locality of the lattice Boltzmann method is conserved, this boundary treatment method is best suited for the use of GPUs in the calculation of the flow field in the reactor. If the mass loss problem is solved with this type of boundary treatment is subject to further research. The Smagorinsky model is used to model the subgrid turbulence (Smagorinsky, 1963; Yu et al., 2005).

2.2.2 Bubble movement

To model the huge amount of bubbles in a reactor a parcel approach was used. Bubbles with similar properties are grouped together. Every action like the movement or a breakup is only done within a parcel and not for each bubble in the parcel. To capture the bubble movement the forces acting on the bubbles like stress gradients, net gravity, drag, lift, and added mass are calculated using the reduced equations of motion (Hu and Celik, 2008):

$$\frac{dv}{dt} = -2g + 3\frac{Du}{Dt} - \frac{3}{2d_b} c_D |v-u|(v-u) - 2c_L(v-u)\times\omega \tag{6}$$

Variables like the fluid velocity are mapped to the bubble location with a 4th-order mapping function (Deen et al., 2002):

$$\zeta(x-x_p) = \frac{15}{16}\left[\frac{(x-x_p)^4}{n^5} - 2\frac{(x-x_p)^2}{n^3} + \frac{1}{n}\right] \text{ with } |x-x_p| \le n \tag{7}$$

Then, the particle fluctuation velocity is calculated using the formulation of Sommerfeld (Sommerfeld et al., 1993). The drag coefficient c_D is estimated following the formula of Tomiyama (Tomiyama, 1998) for fully contaminated systems:

$$c_D = \max\left[\frac{24}{\mathrm{Re}}\left(1+0.15\,\mathrm{Re}^{0.687}\right), \frac{8}{3}\frac{Eo}{Eo+4}\right] \tag{8}$$

The Reynolds number was corrected following the formulation of Han (Luchang et al., 2009) to include the effect of the turbulent fluctuation:

$$\mathrm{Re} = \frac{\rho d_b |u-v|}{\mu} \tag{9}$$

$$\mu = \mu_l + C_l \rho \varepsilon^{1/3} d_b^{4/3} \text{ with } C_l = 0.02 \tag{10}$$

The lift force coefficient c_L is calculated using the formulation of Darmana (Darmana et al., 2006). The forces are then used to update the velocities of the bubbles.

The collisions are modelled using the stochastic inter-particle collision model (Sommerfeld, 2003). The contact of the current bubble with a fictional collision partner is determined with the collision probability based on local statistics. The point of impact on the bubble surface is randomly determined on a collision cylinder, and the collision is assumed to be inelastic with a predefined coefficient of restitution. Following the bubble collision, coalescence of the bubble is determined by comparing the contact time and the film drainage time. Coalescence takes place when the contact time is longer than the film drainage time.

The bubble breakup model of Luo and Svendsen (Luo and Svendsen, 1996) which is used in this study assumes that the turbulent fluctuations are dominating the breakup mechanism and only eddies of a size smaller than or equal to the bubble diameter cause

bubble breakup. If breakup occurrs, the number of bubbles in the parcel is increased while the bubble diameter decreases. The overall volume of the parcel stays the same.

For the backward coupling the forces which the bubbles exhibit on the fluid are calculated. The sum of the drag, the lift and the buoyancy force are distributed to the fluid nodes with the 4^{th} order mapping function.

2.2.3 Graphic processing units

The programs, which are executed on the GPU are called kernels. A copy of the kernel is executed in one thread, which is the smallest unit to be calculated on a GPU multiprocessor core. Every calculation for one node is executed in a separate thread.. A group of threads forms a block. The blocks can be addressed as a two-dimensional array with an x- and a y-value. The simulation area is therefore divided into threads in x-direction and into a block array in the y- and z-direction.

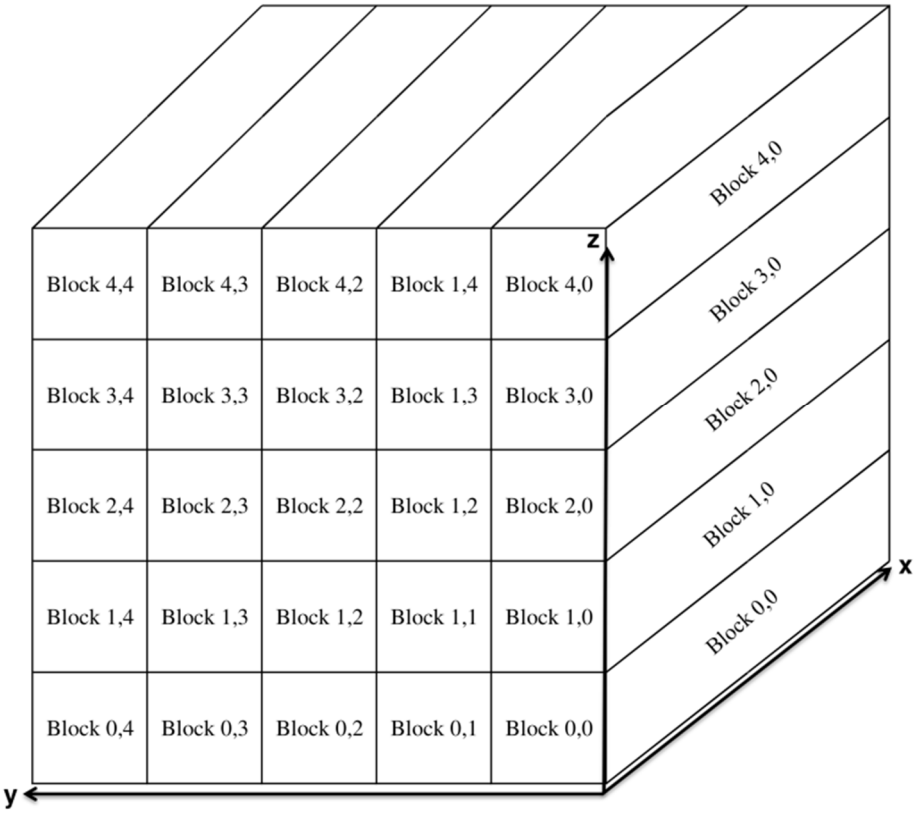

Figure 1: Allocation of the simulation area

The kernel is written for one thread. The identification number of the thread in the block, together with the block coordinate, give the spatial location of the calculation node, which is then used to determine at which position of the respective arrays the input variables should be read and the output variables should be stored.

The biggest performance limiting factor when using CUDA is the memory access. Every thread obviously needs the input values for the calculation and reads randomly from the memory in the worst case. Therefore, the values have to be stored regularly, which means that for example the variable A for thread 1 is located next to the variable A of thread 2 and so forth. This is called aligned memory access and is a requirement to achieve the promised parallelization speedup. An ideal calculation algorithm should

therefore not require random read and write operations. This is almost fulfilled by the lattice Boltzmann method (Tölke, 2008).

To achieve double precision accuracy with single precision variable lengths the fluid density times the weighting factor is subtracted from the distribution functions (Valderhaug, 2011).

2.3 Results and Discussion

2.3.1 Single phase flow field

To validate the simulation results the PIV data of Montante (Montante et al., 2007) are used. Their reactor is equipped with four baffles and a Rushton turbine with six blades. The simulation parameters are listed in table 1.

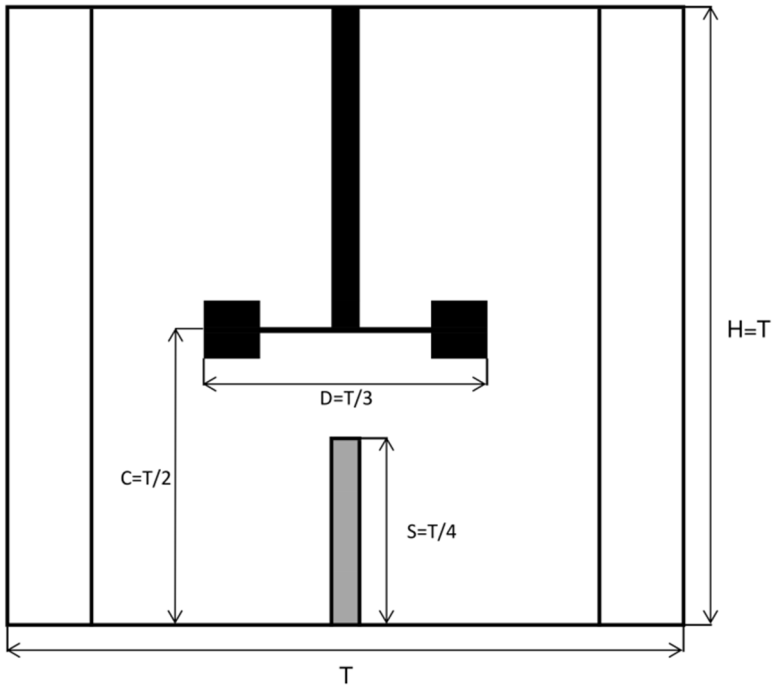

Figure 2: Experimental setup for the Montante test case

Table 1: Simulation parameter of the Montante reactor

Fluid density	1000 kg/m³
Reactor volume	0.01 m³
Impeller diameter	0.078 m
Reactor diameter	0.236 m
Impeller rot. speed	450 rpm

The fluid velocity was averaged after the simulation (Figs. 3-4). The data for the validation were taken after 500.000 time steps (7.7 s in real time). Slight deviations can be seen in the peaks of the axial and radial velocities, but the overall agreement in velocity magnitude is quite good.

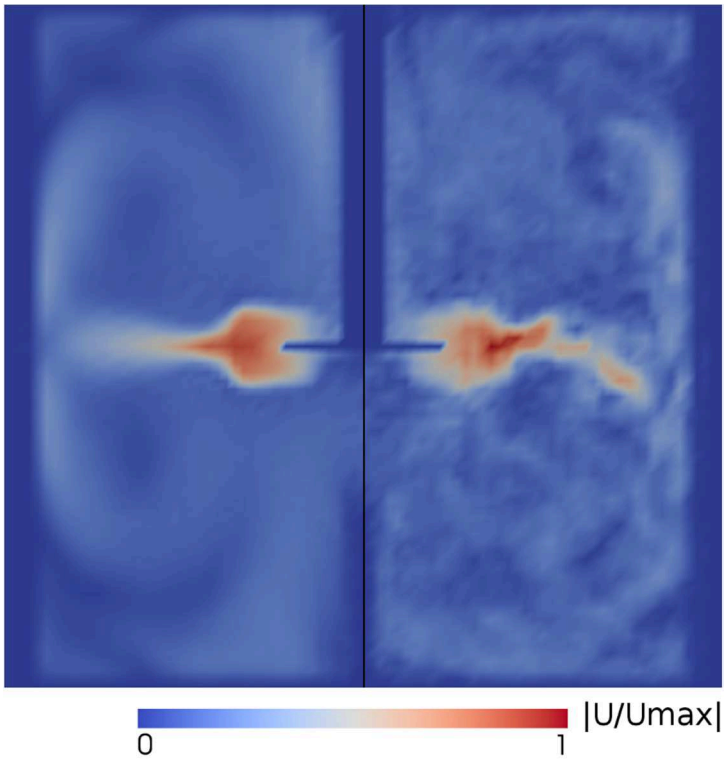

Figure 3: Flow field of a slice though the reactor, left: averaged flow field, right: instantaneous velocities

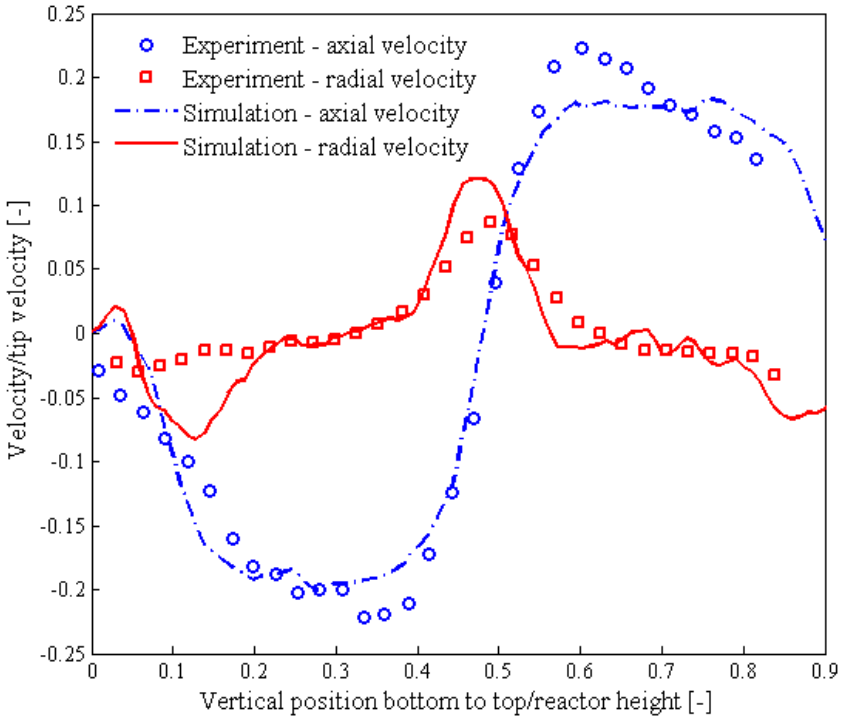

Figure 4: Experimental and predicted axial and radial liquid velocities at r/T=0.46

2.3.2 Swirling flow validation case

To validate the movement of the bubbles, a test case is needed, where the physically correct movement of the bubbles can be calculated. In the swirling flow test case (Figure 5) the bubble motion in a flow that rotates with a constant angular velocity and a uniform axial velocity is studied (Magaud, 2003). The added mass, lift, drag, fluid stress and buoyancy forces are taken into account. The analytical solution is in the case of a rotational speed of 522 rpm (Hu, 2005):

$$r^*(t^*) = r_0^* \exp\left[-f \cdot t^*/2\right]\left(\cos\left(\frac{\sqrt{\Delta}}{2}t^*\right) + \frac{f}{\sqrt{\Delta}}\sin\left(\frac{\sqrt{\Delta}}{2}t^*\right)\right) \text{ with } r^* = \frac{r}{R}, \quad t^* = \omega \cdot t \tag{11}$$

$$f = \frac{3k}{8}\frac{\mu/\rho}{\omega(d_b/2)^2} \text{ and } \Delta = \left|f^2 - 8\right| \tag{12}$$

Table 2: Simulation parameter for the swirling flow case

Fluid density ρ	1000 kg/m³
Bubble diameter d_b	0.001 m
Bubble initial velocity $v_{b,0}$	1.312 m/s
Fluid velocity in z-dir. w_0	1.0 m/s
Fluid rotational speed ω	54,66 rad/s
Bubble release point $x_{b,0}$	(0.024m, 0m, 0m)
Air density ρ_b	1.0 kg/m³
Constant k	24
Dimensions (x, y, z)	0.06m, 0.06m, 1.2m
Tube radius R	0.03 m
Viscosity μ	0.001 Pa s
Drag coefficient c_D	24/Re
Lift and added mass coeff.	0.5

Stoke's drag law is in principle not valid for the simulation parameters. However, the use of this law is necessary to use the analytical solution for validation. The simulation showed good agreement with the analytical solution (Figure 6).

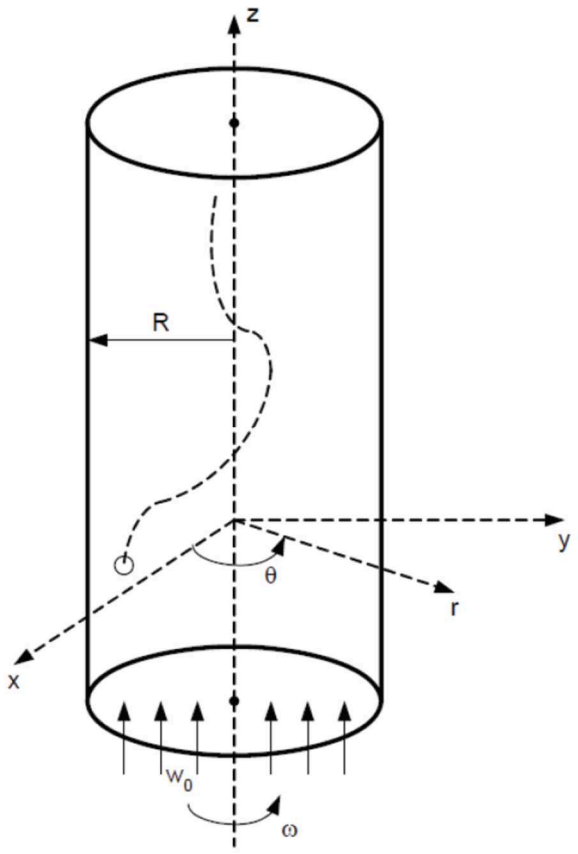

Figure 5: Setup of the swirling flow test case (Radl, 2010)

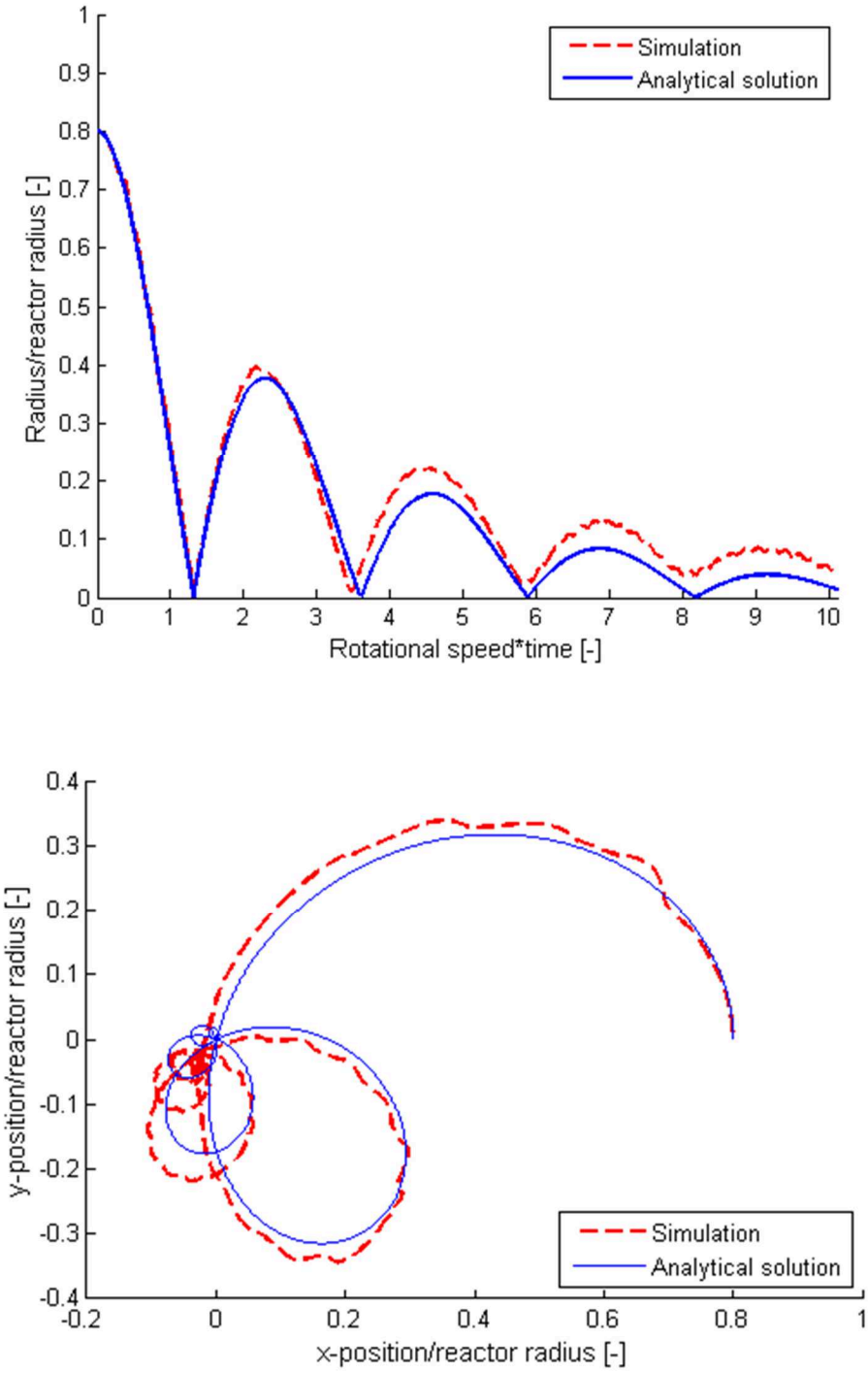

Figure 6: Comparison of the simulation and the analytical solution – top: radial distance of the bubble position to the impeller axis over time; bottom: top view of the bubble movement

2.3.3 Two phase reactor simulation

For the two-phase flow simulation the Montante PIV results (Montante et al., 2007) were used. The simulations were done with a gas flow rate of 0.02 vvm and a stirrer speed of 450 rev/min. The initial gas bubble diameter was fixed with 4 mm. The long term averaged fluid and gas velocities are compared in figure 7.

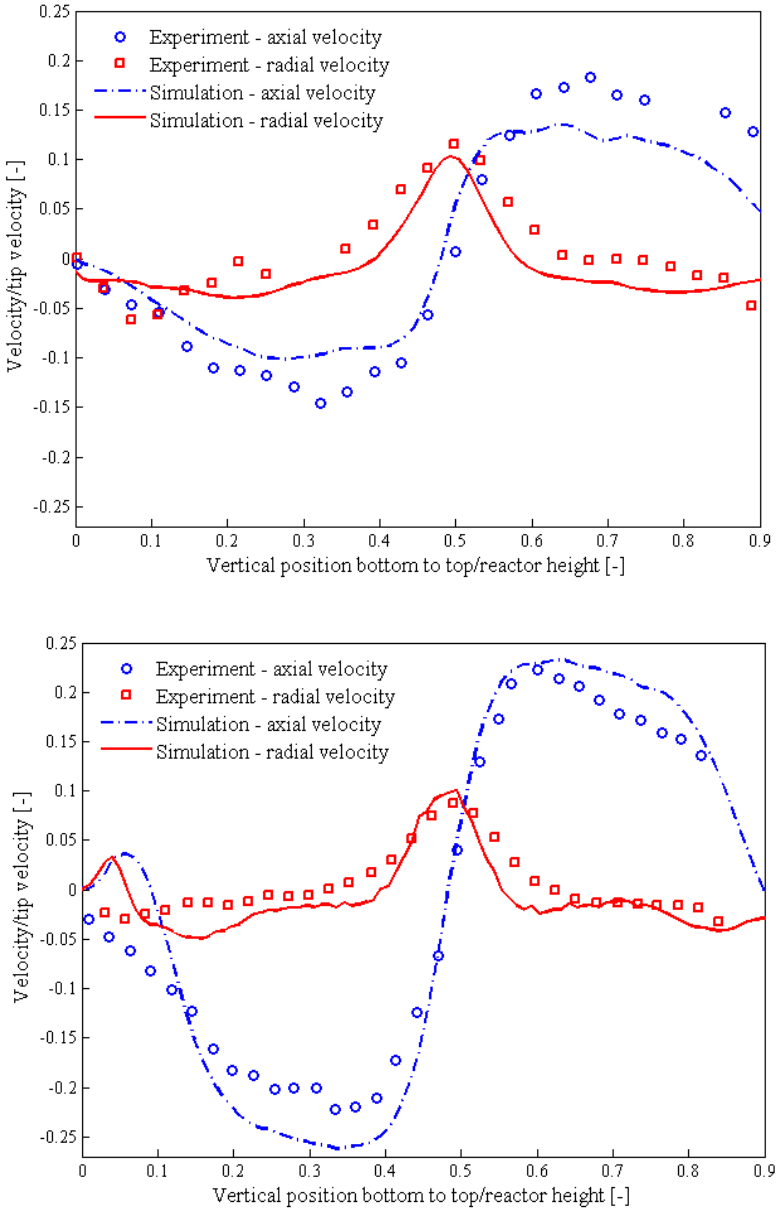

Figure 7: Experimental and predicted axial and radial bubble (top) and liquid (bottom) velocities at r/T=0.46

Good agreement was reached between the experimental and the simulation results. Subject to further research is the validation with holdup values and bubble sizes at various operating parameter in Chapter 3.

2.3.4 Multi-GPU

To speed up the simulation the workload is distributed to two graphic processing units. To communicate the results of the processors after each time step data fields are copied from the memory of the GPU which is calculating the LBM part to GPU which is doing the LPT calculation and vice versa. Several possibilities exist for this transfer. For a high efficiency two buffers are created per data field. With a flip flop scheme one of the buffers is utilized for storing the new values during the calculation. The other buffer with the values of the last time step is copied to the other GPU while both GPUs are working on the current time step. Although this scheme is the fastest because of the co-current copy process, it showed deviating results in the swirling flow test case and is also consuming memory space on the graphic card, which is a very limited resource.

Another way to exchange the data between the processors is to wait until both, the LBM and the LPT calculation kernels are finished and then copy the data fields without the use of buffers. This saves memory space and shows good results (see figure 8) in the swirling flow validation case (see table 3) but is not as fast as the method with buffers.

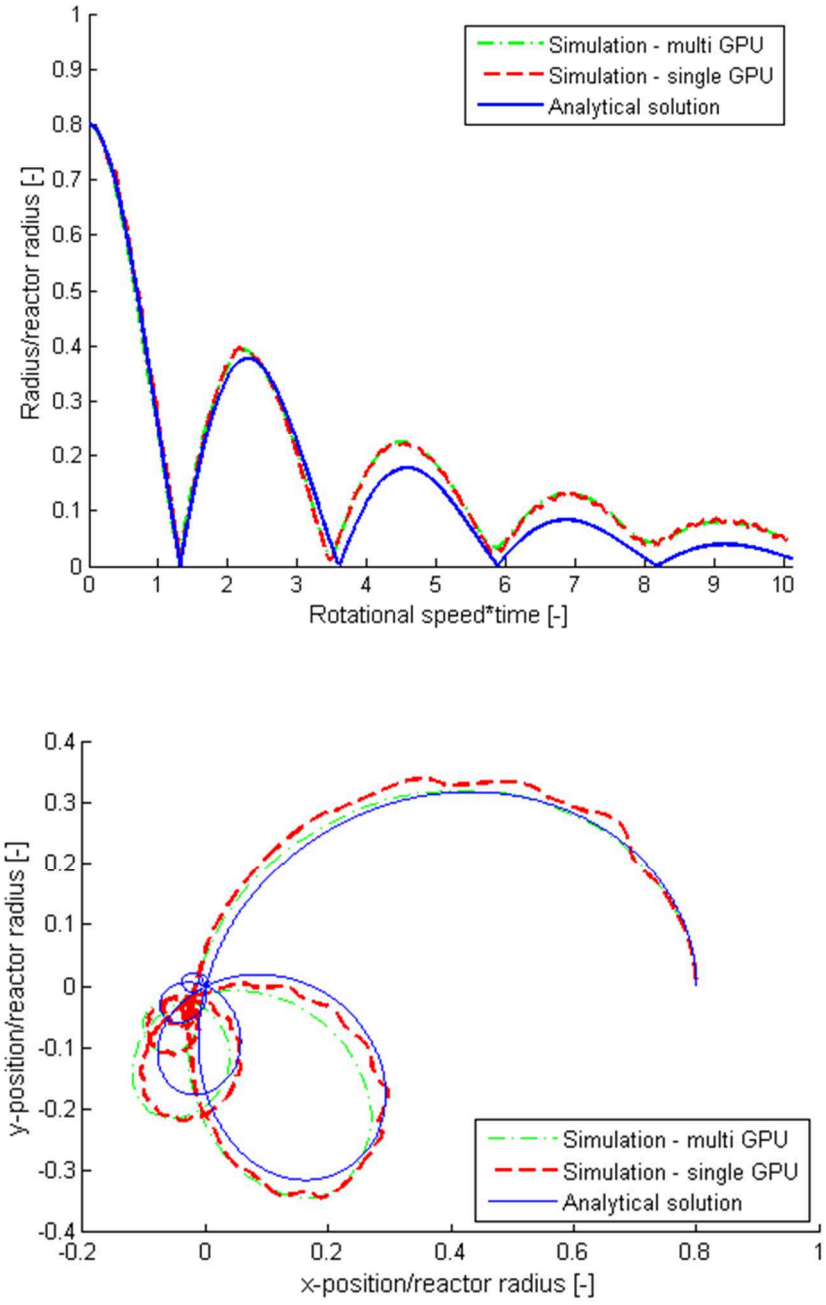

Figure 8: Comparison single - multi GPU –top: radial distance of the bubble position to the impeller axis over time; bottom: top view of the bubble movement

2.3.5 GPU timing

For the determination of the speed up, the GPU code running on a Nvidia GeForce GTX 590 with 1024 computing units was compared to the CPU code running on the iCluster at the TU Graz. The time required for 50,000 time steps in the Montante reactor

simulation was measured with the CPU code running on one CPU, three CPUs and on nine CPUs. This was then compared to the execution time of the same amount of time steps on the GPU. Compared to a single CPU the GPU is fifty times faster for the LPT and LBM calculation. Even when nine CPUs are used, the GPU is thirty times faster (see table 3).

Table 3: Speed up of the simulation running on GPU compared to the CPU to reach 50.000 time steps

Single CPU	7.40 h
Three CPUs	7.34 h
Nine CPUs	4.79 h
Single GPU	2.47 h
Dual GPU	0.15 h

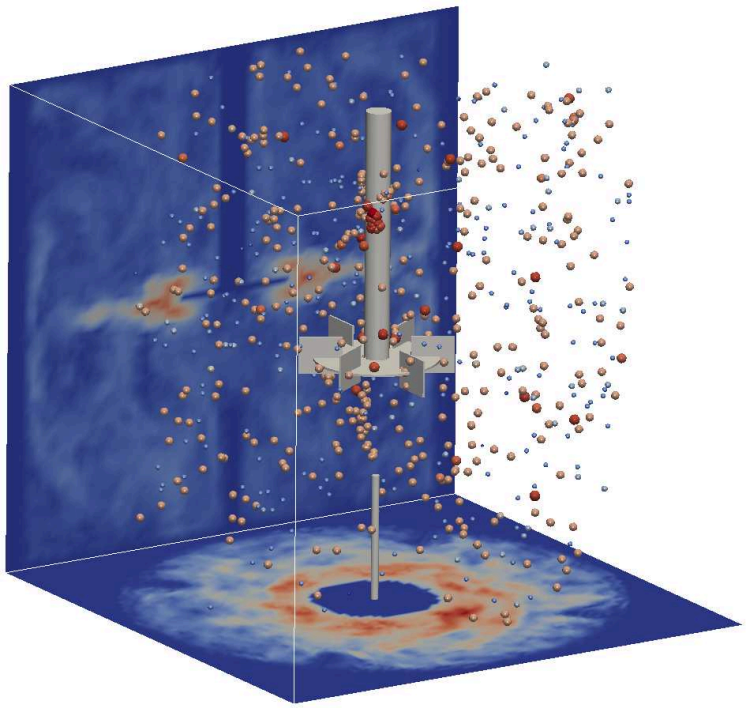

Figure 9: Simulation result of the Montante reactor with Rushton turbine and gas sparger

This clearly shows that the implementation is very efficient on the GPU and that the speed up compared to the CPU version is significant. The speed up is even more

remarkable if the fact is taken into account that the CPU version was running on a scientific cluster compared to a normal desktop machine with a modern graphic card which the GPU code was running on.

2.4 Conclusions

A gas-liquid two-phase flow reactor simulation technique for large systems has been introduced. The fluid field is modelled using the lattice Boltzmann method with 19 statistical distribution functions around each node. The turbulence is calculated using a Smagorinsky subgrid model. The bubble movement was captured based on Lagrangian particle tracking. Bubbles with similar properties are grouped to a parcel. The movement of the parcel is calculated using the reduced equations of motion. Collisions between bubbles are modelled with a stochastic inter-particle collision model based on the local statistics. Coalescence, as well as breakup, are determined based on the contact time and the turbulent fluctuation, respectively. The validation was done with the Montante reactor and the swirling flow case.

The calculations are performed extraordinarily efficient on a graphic processing unit as the algorithms are almost entirely based on the information stored on the calculation node. Therefore, an acceleration of the simulation by a factor of thirty could be achieved compared to the CPU code running on nine processors on a scientific cluster.

2.5 Nomenclature

C – stirrer height from bottom, m

c_D – drag force coefficient

c_L – lift force coefficient

C_l – turbulence correction constant

d_b – bubble diameter, m

D – stirrer diameter, m

Eo – Eötvös number

e_a – spatial direction

ε – turbulent energy dissipation, m²/s³

f_a^{eq} – equilibrium function

f_a – statistical distribution function

$f_{a'}$ – statistical distribution function after the bounce back

F – force, N

g – gravitational acceleration, m/s²

H – reactor height, m

μ – viscosity, Pa·s

n – influence radius (mapping function, $1.5\,d_b$), m

ω – fluid rotational speed, rad/s

ρ_b – air density, kg/m³

ρ – fluid density, kg/m³

r_0^* – bubble radius at release

R – tube radius, m

r^* – dimensionless radius

Re – Reynolds number

S – gas sparger height, m

t^* – dimensionless time

T – reactor diameter, m

τ – relaxation factor

t – time, s

u – fluid velocity, m/s

u_b – solid part velocity, m/s

v – bubble velocity, m/s

$v_{b,0}$ – bubble initial velocity, m/s

w_0 – fluid velocity in z-direction, m/s

w_a – specific weighting factor

$x_{b,0}$ – bubble release point, m

x – fluid node position (mapping function), m

x_p – bubble position (mapping function), m

2.6 References

Arlov D, Revstedt J, Fuchs L. 2008. Numerical simulation of a gas–liquid Rushton stirred reactor–LES and LPT. *Comput. Fluids* **37**:793–801.

Darmana D, Deen NG, Kuipers J a. M. 2006. Detailed 3D Modeling of Mass Transfer Processes in Two-Phase Flows with Dynamic Interfaces. *Chem. Eng. Technol.* **29**:1027–1033.

Deen NG, Solberg T, Hjertager BH. 2002. Flow generated by an aerated Rushton impeller: two-phase PIV experiments and numerical simulations. *Can. J. Chem. Eng.* **80**:1–15.

Derksen JJ, van den Akker HEA. 1999. Large eddy simulations on the flow driven by a Rushton turbine. *AIChE J.* **45**:209–221.

Eggels J. 1996. Direct and large-eddy simulation of turbulent fluid flow using the lattice-Boltzmann scheme. *Int. J. heat fluid flow* **17**:307–323.

Goldstein D, Handler R, Sirovich L. 1993. Modeling a no-slip flow boundary with an

external force field. *J. Comput. Phys.* **105**:354–366.

Guo Z, Zheng C, Shi B. 2002. An extrapolation method for boundary conditions in lattice Boltzmann method. *Phys. Fluids* **14**:2007–2010.

Han L, Liu Y, Luo H. 2007. Numerical simulation of gas holdup distribution in a standard Rushton stirred tank using discrete particle method. *Chinese J. Chem. Eng.* **15**:808–813.

Hu G. 2005. Towards Large Eddy Simulation of dispersed gas-liquid two-phase turbulent flows. ; West Virginia University, Morgantown, U.S.

Hu G, Celik I. 2008. Eulerian–Lagrangian based large-eddy simulation of a partially aerated flat bubble column. *Chem. Eng. Sci.* **63**:253–271.

Ladd A. 1994. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *J. Fluid Mech.* **211**:285–309.

Luchang H, Yuejin L, He'an L, Yang C, Ge B. 2009. Numerical study of the effect of eddy-bubble interaction model on the turbulent dispersed flow of gas bubbles in stirred tanks. *Sci. Online* **278**:1–14.

Luo H, Svendsen HF. 1996. Theoretical Model for Drop and Bubble Breakup in Turbulent Dispersions. *AIChE J.* **42**:1225–1233.

Magaud F. 2003. Modeling and qualitative experiments on swirling bubbly flows: Single bubble with Rossby number of order 1. *J. Fluids Eng.* **125**:239–246.

Montante G, Paglianti a, Magelli F. 2007. Experimental Analysis and Computational Modelling of Gas–Liquid Stirred Vessels. *Chem. Eng. Res. Des.* **85**:647–653.

Obrecht C, Kuznik F, Tourancheau B, Roux J-J. 2011. A new approach to the lattice Boltzmann method for graphics processing units. *Comput. Math. with Appl.* **61**:3628–3638.

Obrecht C, Kuznik F, Tourancheau B, Roux J-J. 2013. Multi-GPU implementation of the lattice Boltzmann method. *Comput. Math. with Appl.* **65**:252–261.

Qian Y, D'Humières D, Lallemand P. 1992. Lattice BGK models for Navier-Stokes equation. *EPL (Europhysics Lett.)* **17**:479–484.

Radl S. 2010. Modeling of multiphase systems in pharmaceutical applications. ; Graz University of Technology.

Smagorinsky J. 1963. General circulation experiments with the primitive equations: I. The basic experiment. *Mon. Weather Rev.* **91**:99–164.

Sommerfeld M. 2003. Euler/Lagrange calculations of bubbly flows with consideration of bubble coalescence. *Can. J. Chem. Eng.* **81**:508–518.

Sommerfeld M, Kohnen G, Rüger M. 1993. Some open questions and inconsistencies of Lagrangian particle dispersion models. *Proc. Ninth Symp. Turbul. Shear Flows*:Paper No. 15-1.

Stobiac V, Tanguy P a., Bertrand F. 2013. Boundary conditions for the lattice Boltzmann method in the case of viscous mixing flows. *Comput. Fluids* **73**:145–161.

Sukop M, Thorne D. 2006. Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers. . Berlin, Heidelberg: Springer.

Sungkorn R, Derksen JJ, Khinast JG. 2011. Modeling of turbulent gas–liquid bubbly flows using stochastic Lagrangian model and lattice-Boltzmann scheme. *Chem. Eng. Sci.* **66**:2745–2757.

Sungkorn R, Derksen JJ, Khinast JG. 2012a. Euler–Lagrange modeling of a gas–liquid stirred reactor with consideration of bubble breakage and coalescence. *AIChE J.* **58**:1356–1370.

Sungkorn R, Derksen JJ, Khinast JG. 2012b. Modeling of aerated stirred tanks with shear-thinning power law liquids. *Int. J. Heat Fluid Flow* **36**:153–166.

Tölke J. 2008. Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA. *Comput. Vis. Sci.* **13**:29–39.

Tomiyama A. 1998. Struggle with computational bubble dynamics. *Multiph. Sci. Technol.* **10**:369–405.

Valderhaug T. 2011. The Lattice Boltzmann Simulation on Multi-GPU Systems. ; Norwegian University of Science and Technology.

Wu J, Shu C. 2009. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *J. Comput. Phys.* **228**:1963–1979.

Yu H, Girimaji SS, Luo L-S. 2005. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *J. Comput. Phys.* **209**:599–616.

3

Local Gas Holdup Simulation and Validation of Industrial-scale Aerated Bioreactors[1]

To date, the efficiency of industrial-size bioreactors has mainly been improved based on empirical knowledge. Computer simulation may help to understand the processes that occur inside the reactor and to develop new reactor designs. Euler-Lagrange simulations of the two-phase flow in large bioreactors, which could not be performed within a timeframe suitable for engineering purposes due to the limited computation resources, were made possible by the calculation power of graphic cards. The lattice Boltzmann method is well suited for parallelization which makes it ideal for calculating the fluid field inside a reactor driven by multiple Rushton turbines on graphic processing units. The bubble movements were captured via a Lagrangian approach by solving the Newton's equations of motion. A two-way coupling between the disperse and continuous phases was applied. Break up and coalescence of the bubbles were modeled via stochastic algorithms using the approach rate of small turbulent eddies and the comparison of the contact time and film breakage time, respectively. To gather experimental data, a conductivity sensor was used to measure the local gas holdup. The rate and the duration of current drops were recorded to estimate the bubble size and the void fraction around the sensor's tip position. The sensor was used in a 150l custom-built acrylic reactor. Several flow regimes with varying gas flow rates and stirrer speeds were investigated. The experimental results were in good agreement with the simulation data, especially at low stirring and low aeration rates. To prove the applicability of the code to large-scale problems, a 40 m³ reactor was simulated.

[1] This chapter is based on Witz C, Treffer D, Hardiman T, Khinast J. 2016. Local gas holdup simulation and validation of industrial-scale aerated bioreactors. Chem. Eng. Sci. 152:636–648.

3.1 Introduction

Bioreactors, also known as fermentors, are widely used in various industrial sectors, including the (bio-)pharmaceutical industry for the production of modern drugs, such as anti-infectives, monoclonal antibodies and other protein drugs, e. g. EPO or insulin (Chu and Robinson, 2001; Jordan, 1995; Nikolai and Hu, 1992; Warnock and Al-Rubeai, 2006). Both small and large-molecule drug substances are produced in bioreactor fermentations. Reactor scales range between a few hundred milliliters on the laboratory scale to several hundreds of cubic-meters at full production scale of microbial cultures. Full-scale production in cell cultures is usually carried out in smaller systems up to a few cubic-meters. Different types of bioreactors are in use today, ranging from wave reactors, shaker bottles, packed beds, airlift reactors, membrane bioreactors to stirred tanks, the latter of which are historically and still the most important systems. The mode of operation is typically fed-batch, although batch and continuous operation are in use as well. Although bioreactors have been used industrially for many decades rational performance optimization remains a challenge (Rani and Rao, 1999; Roubos et al., 1999). The most critical factors in the operation of bioreactors are the (1) gas flow rate applying pressurized air that provides microorganisms or cells with oxygen, the (2) stirrer speed that largely determines the main fluid flow, gas bubble break-up and distribution, and thus, ideally the homogenous distribution of nutrients and products including the exchange with the gas phase, the (3) feed rate of nutrients and pH titration and the (4) heat transfer and thermal control of the reactor. Design factors are the reactor size and geometry, the type, position and number of stirrers (e.g. Rushton, axial pumping impellers, elephant ears, etc.), the shape and

number of heat exchangers (internal vs. external), the level and number of feeding ports, the number and dimensions of baffles and the sparger type and design.

In a production setting, the gas flow rate and the stirring rate can be adjusted based on the actual demands. Often this is done based on empirical knowledge and experience applying correlation equations that are usually established for specific conditions and standard stirrers. However, for the rational design, scale-up, control and optimization of a fermentor more detailed knowledge is highly desired. Typical questions include:

- How does the impeller design impact the overall flow field, mixing time, gas dispersion and hold-up in the reactor?
- Are there inhomogenities and how can they be avoided?
- What shear rates can be expected as a function of impeller design, stirring and gassing rate?
- What is the k_La that can be obtained as a function of the design and operating parameters?
- Which flow regimes are possible and when will stirrer flooding occur?
- How does a non-Newtonian rheology impact the overall mixing pattern in the bioreactor?
- How should nutrients be fed in an optimal way, and what are the associated mixing times?
- How can a smooth scale-up be achieved, i.e., how can one design a system such that microorganisms and cells are exposed to a similar environment on the small, intermediate and production scale?

While it is difficult to answer these questions by experimentation or real-time measurements (at least on large scales), simulation tools that capture the essential

phenomena can provide important insight into the processes, thus providing strategies to rationally optimize the system. However, fully-resolved multi-phase simulations are rarely performed since such simulations still take months to describe only a few seconds of real operation time. This is due to the size of industrial bioreactors and the separation of scales, i.e., small scales in the order of single cells or bubbles co-exist with meso- and macro-scales in the order of the stirrer or reactor diameter (Gillissen and Van den Akker, 2012; Hutmacher and Singh, 2008; Liew et al., 2008).

A recent study on the modeling of gas-liquid stirred tanks was reported by Petitti et al. 2013 who used the Eulerian multi-fluid model to simulate stirred tank reactors including coalescence, breakup and mass transfer of the gas phase. The bubble size distribution inside stirred tank reactors was predicted based on the multiple-size group model by Wang et al., 2014 and with the population-balance models by Morchain et al., 2014. The applicability of various turbulence models was evaluated by Bashiri et al., 2013. Aghbolaghy and Karimi, 2014 analyzed the enzymatic production of hydrogen peroxide and combined the response surface methodology (RSM) and computational fluid dynamics (CFD) to monitor a production process online. However, well-resolved full-scale bioreactor simulations, combining stirring, bubble break-up/coalescence and mass transfer have not been addressed in the literature.

In the simulation of stirred tank bioreactors, the fluid flow field (governed by the Navier-Stokes equations) is resolved via CFD. Numerous algorithms have been developed to approximate the Navier-Stokes equations, such as the finite volume method (Patankar, 1980), the finite element method (Akira, 1986) and the finite difference method (Harlow and Welch, 1965; Richardson, 1911). Another efficient approach, which is ideally suited to be implemented on parallel-computation platforms,

is the Lattice Boltzmann Method (LBM) (Chen and Doolen, 1998a). The LBM is different from other CFD methods as it relies on the collective behavior of groups of "particles" forming the liquid (Yu et al., 2005). Similar to cellular automata, LBM consists of a streaming and a collision step. To enable simulating fluid flows, lattice gas cellular automata were extended by applying a relaxation process (i.e. the system makes a step towards equilibrium) as collision operator instead of a discrete set of collision rules (Wolf-Gladrow, 2000). Bhatnagar, Gross and Krook (BGK) suggested a frequently-used collision procedure and an equilibrium distribution function which acts as objective for the relaxation process (Bhatnagar et al., 1954). Many groups, including the ones of Eggels (Eggels, 1996), Derksen and van den Akker (Derksen et al., 1997; Derksen, 2003; Derksen and van den Akker, 1999; Gillissen and Van den Akker, 2012), Khinast (Sungkorn et al., 2011; Sungkorn et al., 2012a; Sungkorn et al., 2012b) and Bertrand (Stobiac et al., 2013; Stobiac et al., 2014), modeled and simulated stirred tanks with LBM. In the LBM, nodes are evenly distributed throughout the entire domain. At each node, a certain number of spatial probability functions point to the neighboring nodes. Based on the count of the functions, various types of LBM can be distinguished, which are named following a DnQm scheme, where *n* is the dimension of the domain and *m* is the number of probability functions. In our work we applied a D3Q19 model, including a resting function.

An important aspect is the modeling of turbulence, since unsteady vortices of many length scales are present. Multiple models have been developed that describe the effects of those turbulent eddies on the flow field and represent the strongly increased transport of momentum, mass and energy. In the direct numerical simulation (DNS) the whole spectrum of turbulent scales is resolved by solving the Navier-Stokes equations on a

very fine grid (Joshi et al., 2011; Radl and Khinast, 2007). As the size of the mesh and the number of grid points scale higher than quadratically with the turbulent Reynolds number, DNS is not feasible for large-scale bioreactors. However, DNS studies of stirred vessels have been carried out in a Reynolds number range of 1600 to 7300. For example, baffled tanks were simulated by Bartels et al., 2000 and Gillissen and van den Akker, 2009, 2012. Unbaffled tanks were analyzed via DNS by Verzicco et al., 2004, Sbrizzai et al., 2006 and Derksen,2012.

Another way to allow the usage of a coarser grid is the space filtering of the turbulent eddies in the Large-Eddy Simulation (LES). In the LES only eddies larger or equal a certain cut-off size are directly resolved in the simulation. This size is typically the grid spacing. A sub-grid model is used to estimate the contribution of eddies smaller than the grid size. As the small-scale eddies dissipate the turbulent energy of the large eddies the role of the sub-grid model is to remove kinetic energy from the resolved scales (Mathpati and Joshi, 2007). Widely-used models which estimate the contribution of the sub-grid scale stresses include the Smagorinsky model (Smagorinsky, 1963) and the dynamic Smagorinsky model (Germano et al., 1991). The (additive) turbulent or eddy viscosity is calculated with the filtered strain rate tensor and the Smagorinsky constant. Stirred tank simulations via LES and the Smagorinsky sub-grid model were performed by Eggels, 1996, Derksen and van den Akker, 1999, Hartmann et al., 2004a, 2004b, Bakker and Oshinowo, 2004, Yeoh et al., 2004, 2005, Khinast (Sungkorn et al., 2011; Sungkorn et al., 2012a; Sungkorn et al., 2012b), Zhang et al., 2012, Roy and Acharya, 2012, as well as Eng and Rasmuson, 2012. A detailed discussion of some of these studies (Bakker and Oshinowo, 2004; Derksen and van den Akker, 1999; Eggels, 1996;

Hartmann et al., 2004a; Hartmann et al., 2004b; Yeoh et al., 2004; Yeoh et al., 2005) was reported by Joshi et al., 2011.

In the current work, LES is used, since a RANS model with the underlying assumptions (i.e., local isotropy, truncation of higher order terms, etc.) cannot correctly model local transient flow dynamics which are essential for the breakup and coalescence of bubbles in the flow field. A comparison of DNS and LES results by Gillissen and van den Akker, 2012 showed that a Smagorinsky constant of $C_S = 0.1$ is a good approximation throughout the tank. Here, the classical Smagorinsky model with $C_S = 0.1$ is used.

Simulation of the baffles and the rotating stirrer requires special attention. Steady and unsteady approaches can be distinguished. Examples of the steady state approaches include the black box or impeller boundary condition (Harvey and Greaves, 1982a; Harvey and Greaves, 1982b; Kresta and Wood, 1991; Ranade and Joshi, 1990), the source-sink approach (Pericleous and Patel, 1987), the inner-outer approach (Brucato et al., 1998), the multiple-reference frame technique (Deglon and Meyer, 2006) and the snapshot approach (Ranade et al., 2001). One of the objectives of the bioreactor simulation is getting information of the transient behavior. Hence an unsteady-state approach for the boundary handling is used in our work. The sliding-mesh method (Ng et al., 1998) uses a separate rotating grid for the impeller region and a static outer grid which are coupled via a sliding-grid algorithm. The sliding-mesh approach is computationally expensive (Joshi et al., 2011). The moving-deforming grid technique (Perng and Murthy, 1993) uses one grid over the whole domain. This mesh is then deformed as the impeller moves which results in the highest computational demand (Joshi et al., 2011). The LBM requires a regular grid. The advantage of the regular grid is that it requires no customized mesh for different geometries and no special technique

(e.g., the sliding mesh method) for simulating the rotating parts. However, a model has to be developed to treat static and moving boundaries.

Developed for LBM is the modified bounce back rule (Ladd, 1994). Calculation nodes are marked as solid to represent the wall, the stirrer or the baffles. As the grid is regular, curved boundaries lead to a stair shaped geometry. Extrapolation or interpolation methods (Bouzidi et al., 2001; Guo et al., 2002b; Mei et al., 1999) have been developed to improve the treatment of curved boundaries at the price of higher computational demand. The immersed boundary technique (Buick, 2009; Feng and Michaelides, 2004; Wu and Shu, 2009) uses points at the solid liquid boundary to represent the solid parts. At each point a certain fluid velocity is induced by force field acting on the surrounding fluid nodes. We have used the modified bounce back rule to achieve the highest computational efficiency.

Bioreactors are typically multi-phase systems where air bubbles are created at the sparger and rise through the reactor, experiencing coalescence and break-up events. At the same time O_2 is transferred from the gas into the liquid and dissolves (Koynov et al., 2007). Moreover, CO_2 is taken up by the bubbles. The presence of a second phase can be accounted for either by a continuous phase description, i.e., the Euler-Euler method (Krishna et al., 2000; Sokolichin and Eigenberger, 1994; Torvik and Svendsen, 1990; Zhang et al., 2012), or by accounting for individual bubbles (or bubble parcels), i.e., the Euler-Lagrange method (Besbes et al., 2015; Dhotre et al., 2013; Hu and Celik, 2008; Lapin and Lübbert, 1994; Lau et al., 2014; Radl and Khinast, 2010; Webb et al., 1992). In the Lagrangian approach each bubble's exact location is stored and the path is computed by solving the Newton's equations of motion. Rotational energy is usually neglected in bubbles.

Arlov et al., 2008 applied this method and the filtered incompressible Navier-Stokes equations to model a monodisperse bubble suspension in the stirred tank reactor. Derksen, 2003 and Li et al., 2015 used the Euler-Lagrange method to simulate of solid suspensions. Sungkorn et al., 2011, 2012a, 2012b combined Eggels and Somers' LBM (Eggels and Somers, 1995) to calculate the fluid field and the Lagrangian particle tracking technique, including a breakup and coalescence model, to simulate the gas-phase behavior. Bubbles with similar properties were aggregated into parcels and only the parcel's location was tracked. The application of the discrete particle method instead of using bubble parcels as proposed by Han et al. (Han et al., 2007; Han et al., 2010) eliminates the need for statistical models of collision, coalescence and breakup. However, none of the studies presented above combined highly parallelizable methods for the fluid-field calculation, such as the lattice Boltzmann method with accurate polydisperse Lagrangian particle tracking methods on modern highly parallel computation platforms. Only this approach will make the simulations useable for engineers during the bioreactor design process.

Hence the objective of our study was to develop a method to simulate bioreactors (large-scale) using an Euler-Lagrange approach based on LBM and on LES with the Smagorinski model as sub-grid turbulence closure. The code is highly parallelized on graphics processors as described below, thus speeding up the calculation and allowing the simulation of large, industrially-sized stirred bioreactors. In our study, the Bhatnagar-Gross-Krook approximation was used for the collision step under a D3Q19 LBM code. To simulate the bubble movement, the forces acting on the bubble (i.e., drag, lift, buoyancy, history force, added mass effect and gravity) were added up to determine the acceleration at each time step. For collisions, coalescence (Sommerfeld,

2003) and breakup (Luo and Svendsen, 1996), stochastic models were used, which are also local operations, as required for a highly parallelized execution of the code. The number of bubbles in industrial-size reactors is too large (in the order of billions) to be simulated directly, even using the most advanced computational tools. Limiting factors are the memory space and the calculation time. Thus, we followed a parcel approach and grouped bubbles with similar properties together, reducing the memory requirement and eliminating the neighbor search in the collision and coalescence step. Moreover, breakup was performed within a parcel. Whenever the bubble (representing the bubble parcel) breaks, the diameter of the bubbles in the parcel is reduced and the number of bubbles in this parcel is increased. The motion of the parcel was calculated by summing up the forces acting on one bubble that represents the group of bubbles in a parcel. Due to the high volume fraction of the bubbles in industrial-sized reactors, the disperse phase has a significant effect on the fluid flow field. Consequently, the force exerted by gas bubbles on the continuous phase was distributed among the fluid calculation nodes.

In order to use simulation as standard design tool for bioreactors, simulation times have to be reasonably short, i.e., in the order of hours to a few days. While this may be possible for small-scale systems, for industrial-size reactors simulation times are in the orders of months using standard multi-processor workstations to realize sufficient resolution and predictive capability. This is due to the required fine resolution of the flow field, and due to the large number of bubbles in the reactor. Although the simulation may be parallelized by distributing it among several CPUs, the achievable speedup is limited by inter-processors communications. However, when using graphics processor units (GPUs) these problems are circumvented since all of them share the same memory. The Compute Unified Device Architecture (CUDA) technology of

Nvidia takes advantage of both the calculation power of a large number of GPU processors and their common memory. Thus, bubble and particle tracking can be parallelized efficiency. In addition, LBM allows straightforward parallelization as well, as it is a local method. For example, Obrecht et al. (Obrecht et al., 2011; Obrecht et al., 2013b) and Tölke (Tölke, 2008) reported that GPU-based LBM codes significantly accelerate computations. A combination of LBM and Lagrangian particle tracking is ideally suited to take advantage of GPU computing based on CUDA, and should enable reasonable computation times even for industrially-sized reactors.

To parallelize the calculation, the CUDA framework with C++ was used. The simulations were performed on an Nvidia Tesla K40c and a Quadro K6000, with 12 GB global memory and 2880 processors. It took 147.5 h of wall time to complete 800,000 time steps, which represent the first 11.8 seconds of the reactor operation in real time.

To validate the simulations, a 150 l acrylic glass reactor with three Rushton turbines was utilized and experiments were conducted. The local gas holdup was at various positions of the reactor measured at several stirring and aeration rates with a custom-made conductivity needle probe. Computational results were compared with the experimental data. Finally, large-scale simulations of a large 40 m^3 bioreactor are presented as well.

3.2 Materials and Methods

In order to obtain experimental data a baffled 150 l pilot-plant scale bioreactor made of acrylic glass with three Rushton turbines and a ringer sparger was constructed, as depicted in Figure 10. Geometric parameters are provided in Table 4. The reactor wall is

cylindrical with a plain bottom. The baffles, the stirrer disk, the stirrer blades and the shaft are made of stainless steel.

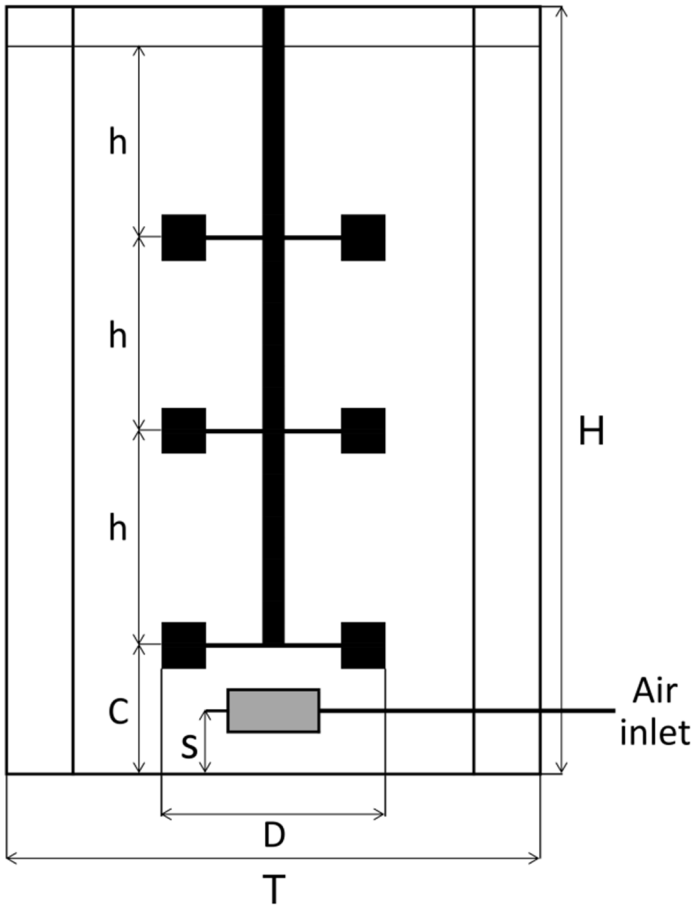

Figure 10: Schematic of the 150l pilot plant bioreactor

The air flow was regulated by a mass flow controller, and air was dispersed by a ring sparger with 36 holes of 2 mm in diameter. Tap water was used in all experiments. Moreover, a holdup sensor based on the work of Bombač (Bombač and Žun, 1997; Bombač and Žun, 2006) was used (Treffer, 2011).

Table 4: Reactor dimensions

Reactor diameter	T	440 mm
Distance between stirrers	h	290 mm
Bottom clearance of lowest stirrer	C	145 mm
Stirrer outer diameter	D	147 mm
Disk diameter		110 mm
Stirrer blade height		30 mm
Stirrer blade width		37 mm
Shaft diameter		50 mm
Air inlet height	s	73 mm

The system has a conductivity needle probe as shown in Figure 11.

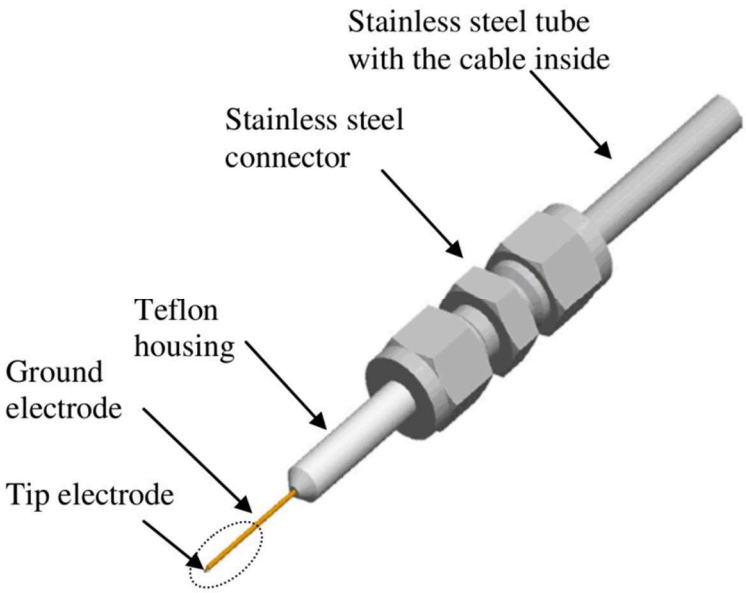

Figure 11: Conductivity probe

An insulated wire inside the needle conducts a current to the tip of the probe. The holdup sensor relies on the difference in the electrical conductivity between water and air. Due to a high conductivity of water, the 1 V, 1 kHz sine signal is conducted between the stainless steel needle and the isolated copper wire. If a gas bubble hits the needle, the circuit is interrupted. Since the interruption time correlates with the bubble size, the local holdup can be measured using a signal processing algorithm in Matlab.

The system was then calibrated using bubble column experiments (Bombač and Žun, 1997).

Table 5: Operating conditions and results of the holdup measurements

	Fr [-]	Fl [-]	N [min^{-1}]	$\dot{G}$ [m³/h]	vvm
F1	0.15	0.02	190	0.71	0.08
F2	0.80	0.02	440	1.64	0.18
F3	1.10	0.05	516	4.82	0.54
F4	0.15	0.05	190	1.78	0.20
F5	0.50	0.12	348	7.79	0.87

The local holdup was measured at 20 positions of the reactor and five operating points that represented five different flow patterns (Table 5).

The operating regimes of stirred aerated tanks are typically defined by the dimensionless Froude (Fr) and Flow (Fl) numbers,

$$Fr = \frac{N^2 D}{g} \tag{13}$$

$$Fl = \frac{\dot{G}}{ND^3} \tag{14}$$

where N is the rotational speed, g the gravitational acceleration, D the diameter of the stirrer and $\dot{G}$ the gas flow rate. Six flow regimes were studied. The operating states used in the experiment and the simulation were selected to represent the various flow regimes (see Figure 12).

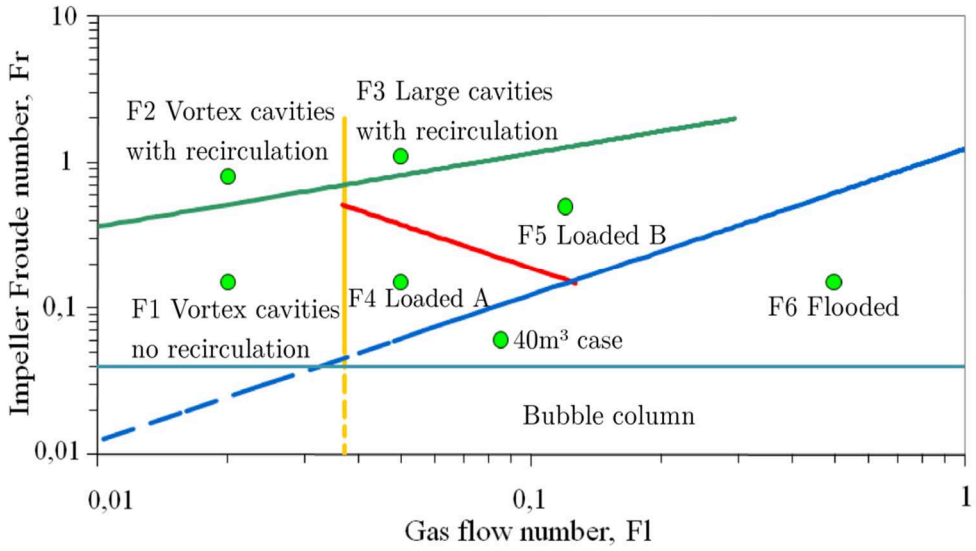

Figure 12: Flow regime map (Treffer, 2011) and location of the simulation/experiments carried out

In F1, the vortex cavities (i.e., a low pressure zone typically behind the impeller blade where gas bubbles accumulate) are visible on the impeller (Treffer, 2011) and the bubbles are not recirculating. Recirculation occurs when bubbles follow the fluid flow to the same stirrer where they just had been radially ejected. The recirculation of bubbles occurs in flow regimes F2 and F3, but with large cavities on the impeller in the latter case. F4 and F5 are examples of the loaded regime with well-dispersed bubbles. In F4, only the lower-most impeller has large cavities, whereas in F5 all impellers have large cavities. F6 denotes the flooded regime where the stirrers, beginning with the lowest, are not able to distribute the gas flow from the sparger. The bubbles then quickly escape to the top of the reactor resulting in a very low gas holdup. F6 was not included in the simulation as the flooded state is generally not desired when operating a bioreactor. Moreover, large bubbles are created which are still hard to simulate with our approach.

To measure the overall holdup the height of the fluid level without air flow and stirring was marked on the reactor wall. After setting the desired operating conditions the increase in fluid height was multiplied with the reactors cross section to obtain holdup values. The results of the experiments were compared to the simulation output. This is reported as "holdup by visual observation" in Table 8.

The experimental and computational local holdup values were compared for the five cases indicated in Table 5 at 4×7 locations. Holdup measurements were done at 0.07 m, which is the level of the sparger, and at 0.17 m, 0.47 m and 0.76 m, which is the height of the stirrers. Moreover, measurements were performed at 0.32 m and 0.61 m, which is midway between the stirrers, and at 0.91 m, which is above the top stirrer. The sensors are positioned on a radial distance to the reactor wall of 2.8 cm, 6.8 cm, 10.5 cm and 13.8 cm. The point with the farthest distance is 0.85 cm apart from the stirrer's outer blade edge. The experimental results are shown in Figure 13 as diamonds. Simulation parameters are provided in Table 6.

Table 6: Simulation parameters

Fluid calculation nodes	$1.13 \cdot 10^6$
Initial bubble diameter	4 mm
Air density	1.20 kg/m³
Water density	1000 kg/m³
Baffle number	4
Rushton impellers	3
Impeller blades	6
Viscosity	0.001 Pa·s
Smagorinsky constant	0.1

To demonstrate the ability of the modeling approach to simulate large reactors, we performed a simulation with 5 million node points corresponding to a reactor volume of 40 m³. The simulation parameters are shown in Table 7.

Table 7: Simulation parameters

Fluid calculation nodes	$4.83 \cdot 10^6$
Stirrer speed	39 rev/min
Reactor height	6.5 m
Reactor diameter	2.8 m
Baffle number	4
Rushton impellers	3
Impeller blades	6
Gas flow rate	0.2 vvm
Froude number	0.057
Flow number	0.089

3.3 Modeling

3.3.1 Liquid flow field

Originating from lattice gas automata (Sukop and Thorne, 2006), the Lattice Boltzmann method (LBM) is an advanced approach for simulating fluid flows. In this method, the nodes are regularly and equidistantly distributed over the domain. Each node is the origin of 19 statistical distribution functions (f_a), each having a different spatial direction (a) with the unit vector (e_a). Due to the orientation of the vectors, a weighting factor (w_a) has to be used. Two steps are required to simulate the fluid flow field which together form Eq. (15). The right hand side of the equation consists of the collision step and the forcing step. In the collision step, the difference between the distribution function and the equilibrium distribution function (f_a^{eq}) is divided by the relaxation

factor τ which is calculated using the fluid's viscosity ν in Eq. (18). This part of the algorithm together with the formulation for the equilibrium distribution function f_a^{eq} (Eq. (17)) is termed the BGK (Bhatnagar-Gross-Krook) approximation (Qian et al., 1992). The forcing term F_a in Eq. (16) adds the body force density $\vec{g}(\vec{x},t)$ using the formulation of Guo (Guo et al., 2002c). The body force density includes the backward coupling force of the bubbles.

$$f_a(\vec{x}+\vec{e}_a\Delta t, t+\Delta t) - f_a(\vec{x},t) = -\frac{f_a(\vec{x},t) - f_a^{eq}(\vec{x},t)}{\tau} + \Delta t F_a(\vec{x},t) \tag{15}$$

$$F_a(\vec{x},t) = \left(1-\frac{1}{2\tau}\right) w_a \left[\frac{\vec{e}_a - \vec{u}}{c_s^2} + \frac{(\vec{e}_a\cdot\vec{u})}{c_s^4}\vec{e}_a\right]\cdot\vec{g}(\vec{x},t) \tag{16}$$

$$f_a^{eq}(\vec{x},t) = w_a\rho\left[1+\frac{\vec{e}_a\cdot\vec{u}}{c_s^2} + \frac{(\vec{e}_a\cdot\vec{u})^2}{2c_s^4} - \frac{\vec{u}^2}{2c_s^2}\right] \tag{17}$$

$$\tau = \frac{\nu}{\Delta t c_s^2} + \frac{1}{2} \tag{18}$$

$$\vec{e}_a = \begin{cases} (0,0,0) & \text{a=0} \\ (\pm1,0,0),(0,\pm1,0),(0,0,\pm1) & \text{a=1-6} \\ (\pm1,\pm1,0),(\pm1,0,\pm1),(0,\pm1,\pm1) & \text{a=7-18} \end{cases} \tag{19}$$

$c_s = (1/\sqrt{3})\Delta x/\Delta t$ is the speed of sound where Δx is the lattice spacing and Δt is the time step. ρ is the density, ν is the kinematic viscosity and $\vec{u}$ the macroscopic fluid velocity. The weighting factors w_a for the D3Q19 lattice are: 1/3 for a=0, 1/18 for a=1-6 and 1/36 for a=7-18. In the streaming step that follows, the relaxed distribution functions are copied to the next node that they are pointing at as shown on the left side of equation

(15). After the streaming step, each node should have a full set of 19 distribution functions again.

The modified bounce-back method's (MBB) (Ladd, 1994) disadvantage is that the reflection of the distribution function occurs halfway between the fluid nodes, creating a stair-step boundary geometry. Yet, if the node number is high enough, this effect is negligible. For the reflection of the distribution function at moving boundaries with the solid part velocity $\vec{u}_b$, the following formula can be used:

$$f_{a'}(\vec{x},t+\Delta t) = f_a(\vec{x},t) - 2w_a\rho\frac{\vec{u}_b\cdot\vec{e}_a}{c_s^2} \tag{20}$$

where $f_{a'}$ is the reflected distribution function pointing in the opposite direction of f_a. This boundary treatment conserves the locality of the LBM and is ideal for graphics processors.

The distribution functions after the two steps are summed up to recover macroscopic fluid properties, such as density and velocity:

$$\rho = \sum f_a(\vec{x},t) \tag{21}$$

$$\rho u = \sum f_a(\vec{x},t)\vec{e}_a \tag{22}$$

Since the information is forwarded to and received from neighboring nodes only in the streaming step, the method is perfectly suited for parallelization via graphics processors, in which read and write operations outside of the current location slow down the calculation.

To simulate sub-grid turbulence, we applied the Smagorinsky model (Smagorinsky, 1963; Yu et al., 2005). The dissipating effect of the turbulent sub-grid eddies is modeled

by increasing the viscosity by the eddy viscosity ν_t. This eddy viscosity is calculated from the momentum flux Q:

$$Q_{ij} = \sum_a \vec{e}_{ai}\vec{e}_{aj}\left(f_a - f_a^{eq}\right) \tag{23}$$

$$Q = \sqrt{2\sum_{i,j} Q_{ij}Q_{ij}} \tag{24}$$

$$\tau_0 = 3\nu_0 + \frac{1}{2} \tag{25}$$

$$\tau_t = \frac{1}{2}\left(\sqrt{\tau_0^2 + 2\sqrt{2}\left(C_{SM}\Delta x\right)^2\left(\rho c_s^4 \Delta t\right)^{-1} Q} - \tau_0\right) \tag{26}$$

$$\nu_t = \frac{1}{3}\tau_t \Delta t \tag{27}$$

$$\nu = \nu_0 + \nu_t \text{ and } \tau = \tau_0 + \tau_t \tag{28}$$

ν_0 is the fluid viscosity and the Smagorinsky constant C_{SM} is set to 0.1 as proposed previously (Gillissen and Van den Akker, 2012; Yu et al., 2005).

3.3.2 Bubble movement

Fluid stresses F_S, gravity F_G, drag F_D, lift F_L and added mass forces F_A are taken into account in the equations of motion (Hu and Celik, 2008):

$$F = F_G + F_S + F_D + F_L + F_A \tag{29}$$

$$F = \left(m_g - m_f\right)\vec{g} + m_f \frac{D\vec{u}}{Dt} - \frac{1}{2} c_D \rho A_{fr} \left|\vec{v} - \vec{u}\right|\left(\vec{v} - \vec{u}\right) \\ - c_L m_g \left(\vec{v} - \vec{u}\right) \times \left(\nabla \times \vec{u}\right) - c_A m_f \frac{d\left(\vec{v} - \vec{u}\right)}{dt} \tag{30}$$

In Eq. (16) m_g is the mass of the bubble, m_f the mass of the fluid of the same volume as the bubble, $\vec{g}$ the gravity, $\vec{u}$ the fluid velocity, $\vec{v}$ the bubble velocity and A_{fr} the cross sectional area of the bubble. c_D denotes the drag coefficient, c_L the lift coefficient and c_A the added mass coefficient. F is replaced by $F = m_g \, d\vec{v}/dt$ and the equation is divided by $\left(m_g + c_A m_f\right)$. The density ratio ρ_g / ρ is neglected as the bubble density ρ_g is small compared to the fluid density. By inserting $c_A = 0.5$ a simple form of the equations of motion is obtained (Hu and Celik, 2008):

$$\frac{dv}{dt} = -2g + 3\frac{Du}{Dt} - \frac{3}{2d_b} c_D \left|v - u\right|(v - u) - 2c_L (v - u) \times \left(\nabla \times \vec{u}\right) \tag{31}$$

d_b is the diameter of the bubble. To interpolate the relevant properties (e.g., fluid velocity), a 4th order mapping function was used (Deen et al., 2002):

$$\zeta\left(x - x_p\right) = \frac{15}{16}\left[\frac{\left(x - x_p\right)^4}{n^5} - 2\frac{\left(x - x_p\right)^2}{n^3} + \frac{1}{n}\right] \text{ with } \left|x - x_p\right| \le n \tag{32}$$

The lift coefficient was calculated according to the formulation reported by Darmana et al., 2006. For the drag coefficient, the formulation of Tomiyama, 1998 for pure systems was applied:

$$c_D = \max\left\{\min\left[\frac{16}{\mathrm{Re}}\left(1+0.15\,\mathrm{Re}^{0.687}\right), \frac{48}{\mathrm{Re}}\right], \frac{8}{3}\frac{Eo}{Eo+4}\right\} \tag{33}$$

Here, the bubble Reynolds and Eötvös number are given by:

$$\mathrm{Re} = \frac{\rho d_b |u-v|}{\mu} \tag{34}$$

$$Eo = \frac{g(\rho-\rho_b)d_b^2}{\sigma} \tag{35}$$

The local statistics around the bubble determine the probability of a collision with a fictitious bubble. The stochastic inter-particle collision model (Sommerfeld, 2003) estimates the point of impact by placing the fictitious bubble randomly inside the collision cylinder. The velocities after the collision are calculated by using the mean velocity of the surrounding bubbles and a random value multiplied by the average deviation of the velocity of the surrounding bubbles. The velocity of the bubble after the collision is then determined by assuming an inelastic impact with a predefined coefficient of restitution. Moreover, the collision time is compared with the drainage time of the liquid film between the colliding bubbles. If the contact time is long enough, the bubble and the fictitious bubble form a new larger bubble, increasing the diameter and decreasing the number of bubbles in the parcel. The bubble breakup is assumed to be caused by turbulent fluctuations with eddy sizes smaller than the bubble diameter. The breakup model of Luo and Svendsen (Luo and Svendsen, 1996) was used to estimate the daughter-bubble size distribution. For the backward coupling force, the drag, lift and buoyancy forces were summed up and distributed to the fluid nodes via the 4th order mapping function described above.

The flow field only experiences the presence of the bubbles by the feedback force. The displacement of the fluid is not considered, which could be achieved by including the volume-average Navier-Stokes equations in the LBM formulation (Darmana et al., 2006). This is, however, not trivial and will be subject of future research as it should have a positive effect on the simulation results especially at high holdup regions.

3.4 Results and discussion

In the first regime F1 the rotational speed of the stirrer is too low to create cavity structures behind the stirrer blades. Additionally, the fluid flow velocity is too slow to create a drag force high enough to recirculate a larger fraction of the bubbles. Therefore, the bubbles, after being radially dispersed and broken up by the lowest stirrer, rise to the axial inflow stream of the middle stirrer. This process is then repeated with the middle and the top stirrer. Without recirculation, the volume below the first stirrer is almost completely free of bubbles excluding the cylinder above the sparger. In the zone above the sparger the largest bubbles can be observed until the bubbles follow the fluid flow to the stirrer blades where they are broken up by the blades and the turbulent eddies created by the movement of the blades. In F2 the fluid is fast enough to induce a drag force which is higher than the buoyancy force and hence a majority of the bubbles hits the same stirrer for several times. This recirculation effect significantly increases the holdup of the reactor. Zones of lower local holdup can be found where the vortices of the stirrers meet. The vortex cavities at the bottom and the middle impeller lead to higher holdup values at the sensor next to the stirrer. The regime F3 also has recirculation and large cavities at the stirrer blades.

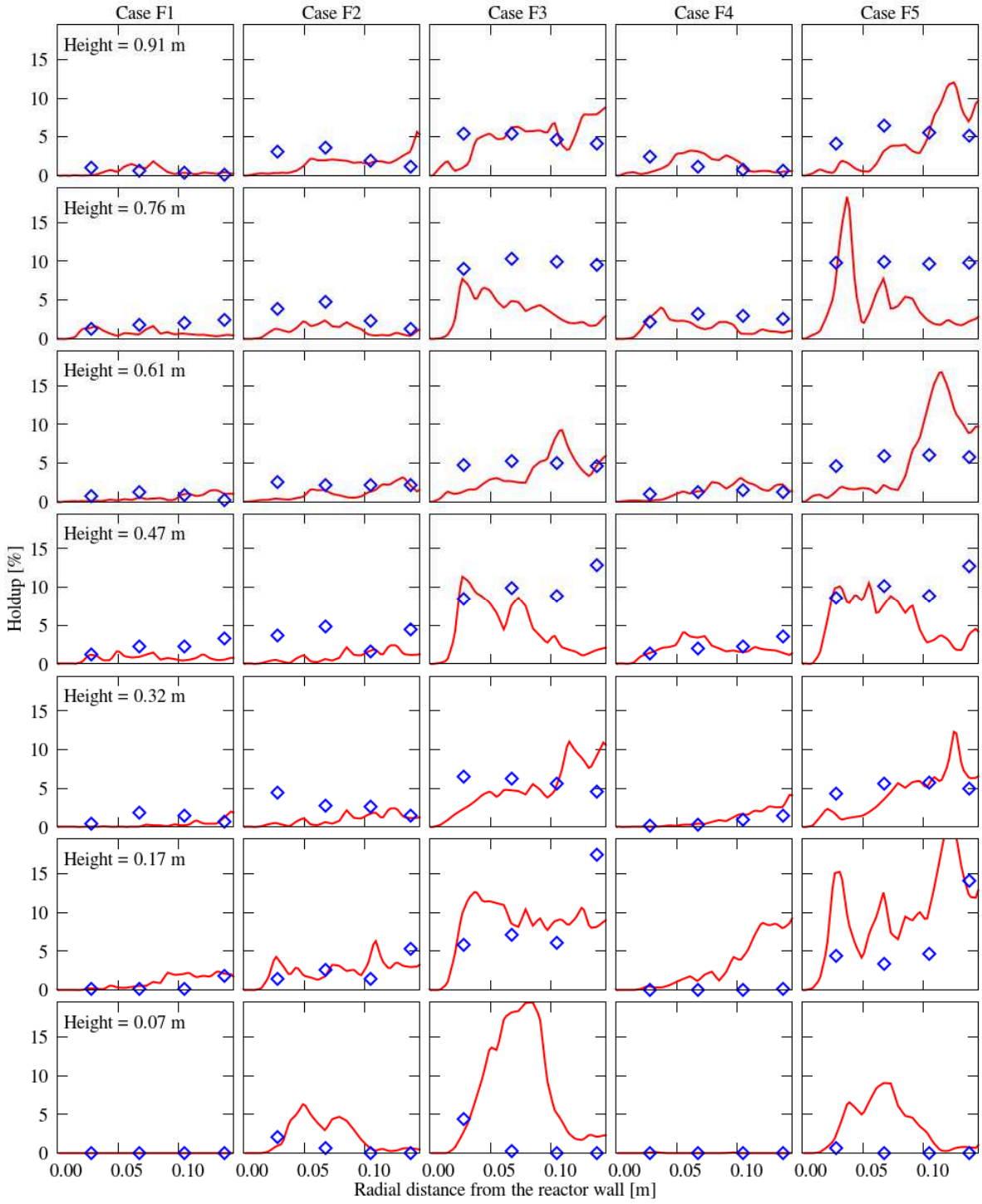

Figure 13: Comparison of the local holdup values. Symbols: Experiment, Lines: Simulation

Again the cavities cause higher holdup values at the sensors next to the blades at the stirrer levels. Interestingly, the recirculation at the lowest stirrer can only be seen at the sensor close to the wall at the sparger level.

In F4 large cavities should appear at the lowermost impeller. In the experiment the bottom impeller showed an oscillating behavior between a flooded state, where it was not able to distribute the bubbles and a state with large cavities. As the flooded time of the stirrer was significantly longer, the measurements were mostly taken during this time. The regime F5 is called “loaded with large cavities on all impellers”. In the holdup diagrams those cavities are visible on the bottom and the middle impeller again with a high value at the sensor closest to the stirrer blades. Again, as in regime F3, the recirculation below the lowest stirrer is only visible by the sensor close to the wall with a lower value than expected.

Simulation results based on our model are shown in Figure 13, as well (solid lines). As can be seen in Figure 13, the holdup values are generally in good agreement (note that our model does not use fitting parameters), except for the sparger level at case F3 and F5, as well as the level of the first stirrer in case F4. The latter can be explained by the dynamics of the lowest stirrer in the F4 case, which alternated slowly between a normal bubble distribution and being flooded, indicating proximity of the flooding regime. In contrast, simulations did not show a flooded impeller.

The overestimation of the holdup values at the sparger level for the cases F3 and F5 may indicate that the value of the drag force in the sparger region is too high. In this case the bubbles are trapped and recirculated in the vortex created by the bottom stirrer which then leads to the overestimation of the holdup in this region. The bubbles that

may create the additional holdup can be seen in Figure 14 cases F3 and F5 at the height level of the sparger close to the center region of the reactor.

Nevertheless, the agreement between experimental and simulation can be considered good given the fact that no fitting parameters are used and that only first principle models and validated closures from literature are applied.

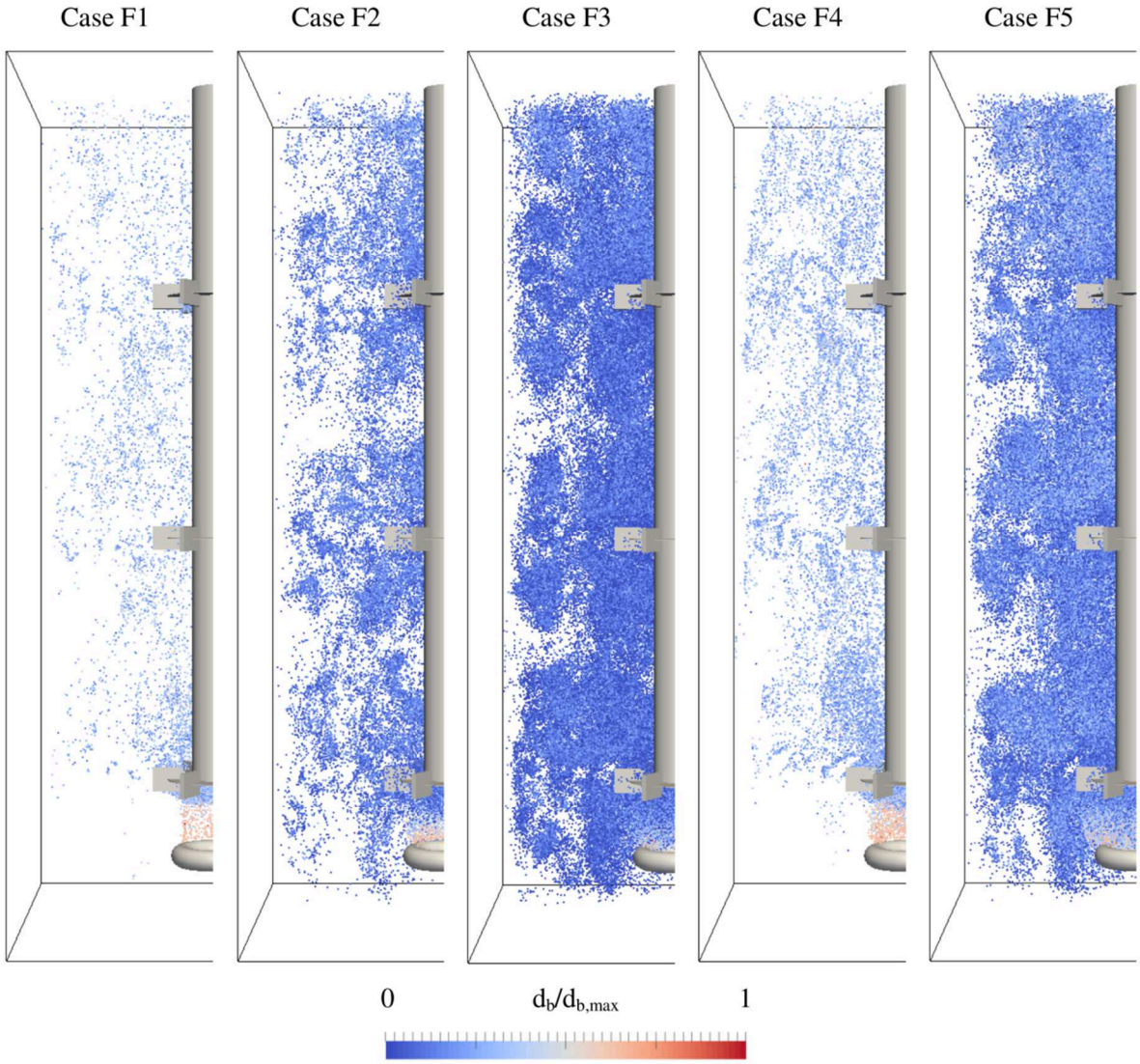

Figure 14: Snapshots of the bubble flow field for the five flow regimes (Table 5, $d_{b,max}$ = 5.97 mm).

Changing the drag model based on the respective case would improve the agreement, yet lessen the applicability and extrapolatory capability of our general large-scale bioreactor simulation model.

Figure 14 shows snapshots of the bubble field for all six cases. The front half of the bubbles has been removed to improve the visibility of the bubble positions at the cross section of the reactor. The real time after the start of gassing of the reactor was 72.2 s for F1, 26.0 s for F2, 22.9 s for F3, 72.2 s for F4 and 14.2 s for F5. At the time of the snapshot in the whole reactor 30,621 bubble parcels are present in case F1, 164,742 in F2, 540,772 in F3, 71,238 in F4 and 510,633 in F5.

The snapshots in Figure 14 reflect the holdup distribution seen in Figure 13. The regimes with recirculation (F2, F3, and F5) display a completely different pattern of the bubble distribution than the regimes without recirculation (F1 and F4). In every regime with recirculation the overestimated holdup at sparger level is visible as bubble cluster near the ring sparger. The dead zones near the wall for the cases with recirculation are visible as empty areas near the wall at half height between the stirrers. In the cases without recirculation these dead zones are larger and start at lower height. In regime F4 the experimental observation that the lowest stirrer is repeatedly being flooded is also visible in the snapshot by the low amount of bubbles radially ejected by the stirrer and the region of high holdup near the stirrer axis.

A comparison of the global holdup values from the simulation after reaching the steady state and the values measured by the height increase of the fluid level are shown in Table 8.

Table 8: Simulation results compared to the holdup measurements

	Holdup visual observation [%]	Simulation result [%]
F1	0.98	0.67
F2	2.93	3.57
F3	8.78	11.75
F4	1.95	1.55
F5	8.29	11.10

As can be seen (and discussed above for the local holdups) the agreement is good in operating points with low gas flow rate and low stirrer speed. For higher gassing rates the predictions of the model are too high in comparison with experiments. However, the trends are quite accurate and allow generating an understanding of the basic process dynamics in a bioreactor.

A slice through the reactor at the mid-plane is shown in Figure 15, highlighting the flow fields, the mass transfer coefficients and the bubble diameters. The figures are standardized to the common maximum of both halves of the respective figure. Figure 15 (a) shows the instantaneous flow velocity at 1.5 million time steps (22.9 s real time) in the left side and the time-averaged flow field on the right side. The snapshot of the flow velocity gives an impression about the highly turbulent flow inside the reactor with the turbulent eddies ejected from the stirrer blades. The average flow field reveals the recirculation cells above and below each stirrer.

Figure 15 (b) depicts the k_La values of case 2 (left) and case 4 (right). Those cases were chosen as they have similar gas flow values (0.18 and 0.20 vvm) but are different in terms of the stirrer speed (440 and 190 rpm). The mass transfer rate for the k_La values was calculated by the penetration theory of Higbie (Higbie, 1935) which was previously used in e.g. (Alves et al., 2004).

$$k_L a = \sum \left(\frac{2}{\sqrt{\pi}} \sqrt{\frac{\vec{u}_{rel} \cdot D_A}{d_b}} A_b \right) \tag{36}$$

D_A is the diffusion coefficient of oxygen, A_b is the surface area of the bubble and $\vec{u}_{rel}$ is the relative velocity between the fluid phase and the bubble. The sum is the sum over all particles in the reactor at the current time step.

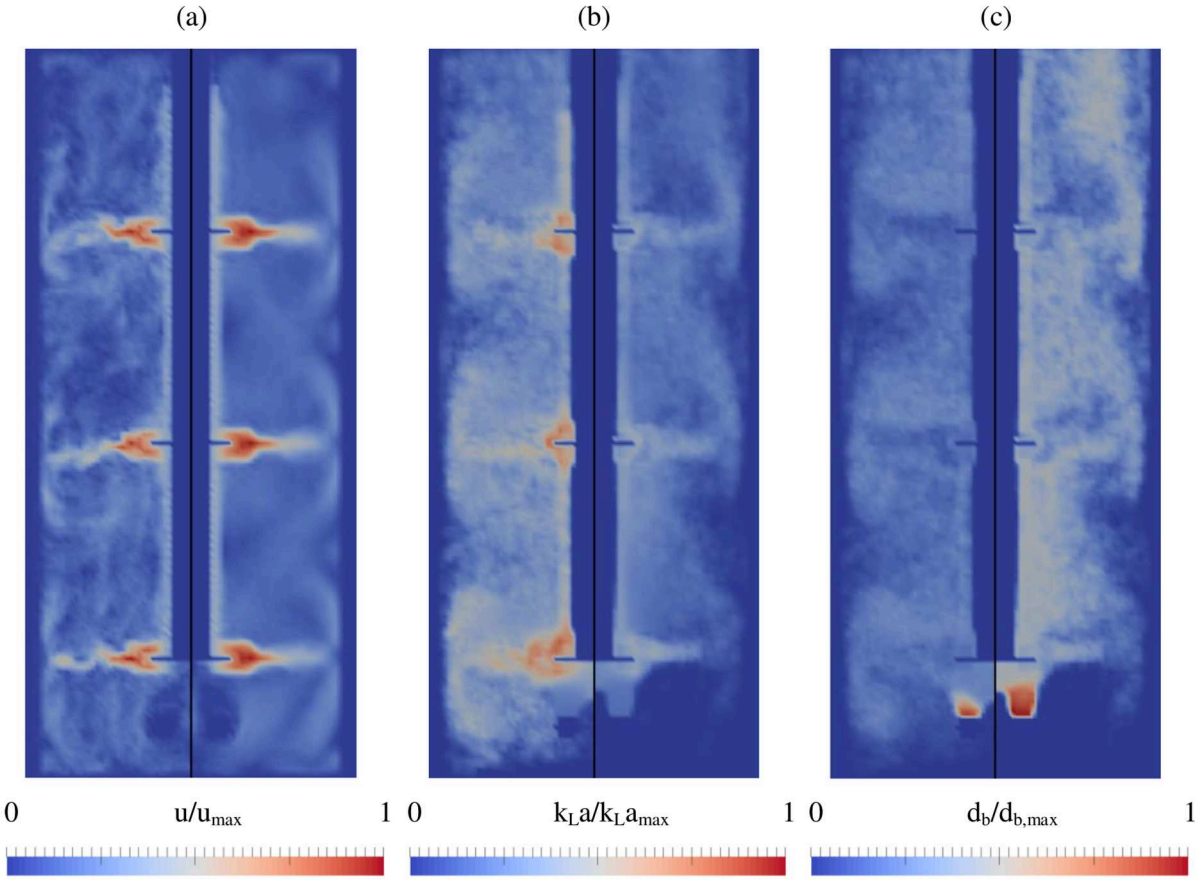

Figure 15 (a) F3 left: fluid flow field snapshot, right: averaged fluid flow field, (b) k_La left: time-averaged F2, right: time-averaged F4, (c)left: time-averaged bubble diameters F2, right: time-averaged bubble diameters F4

The k_La values are also acquired experimentally by removing the dissolved oxygen by stripping with nitrogen (Table 9). After switching to air again the transient

concentration of dissolved oxygen was recorded using an oxygen sensor. The k_La was then calculated using by fitting a function to the recorded oxygen concentration values.

Table 9: Simulation results compared to the k_La measurements

	Holdup visual observation [%]	Experimental result k_La [1/s]	Simulation result k_La [1/s]
F1	0.98	0.0046	0.0023
F2	2.93	0.0188	0.0299
F3	8.78	0.0288	0.1216
F4	1.95	0.0070	0.0060
F5	8.29	0.0182	0.0732

For the cases with higher holdup (F3 and F5) the experimental values are lower as the dissolved oxygen has to be transported to the sensor position at the top of the reactor. As a well-mixed condition is assumed the k_La values are lower than in the simulation where the concentration increase throughout the reactor is instantaneously registered and included in the k_La calculation.

Further insight and a closer approximation of the k_La distribution will be possible with the implementation of a species transport solver.

On the right side of Figure 15 (c) time-averaged bubble diameters are shown. By comparing case F2 on the left side and case F4 on the right, it can be seen that for F2 the average diameter is lower and that the higher fluid velocity (due to higher stirring rates) causes a faster breakup of the bubbles at the sparger. The higher fluid velocity also causes more bubbles to be transported into the area below the lowest stirrer. Combining this information with the mass transfer and a transport equation for oxygen provides valuable insight about the transient oxygen distribution in bioreactors.

Figure 16 shows the integral bubble size distribution for the five cases. As can be seen the bubble size distribution corresponds well with the rotational speed of the stirrer. The regime with the highest rotational speed (F3) shows a left skew in the histogram. With lower stirrer speeds the peak of the size distribution moves to higher bubble diameters. By a comparison of the regimes with equal stirrer speed (F1 and F4) a rather small influence of the gas flow rate (average d_b is slightly smaller in case of F4) can be observed as the distributions only show small deviations concerning the peak and the shape. The high value in the class above 1.890 mm bubble diameter in the regimes F1 and F4 can be explained with the low total number of bubbles in the reactor hence the effect of the large bubbles emitted by the sparger are more observable than in other regimes.

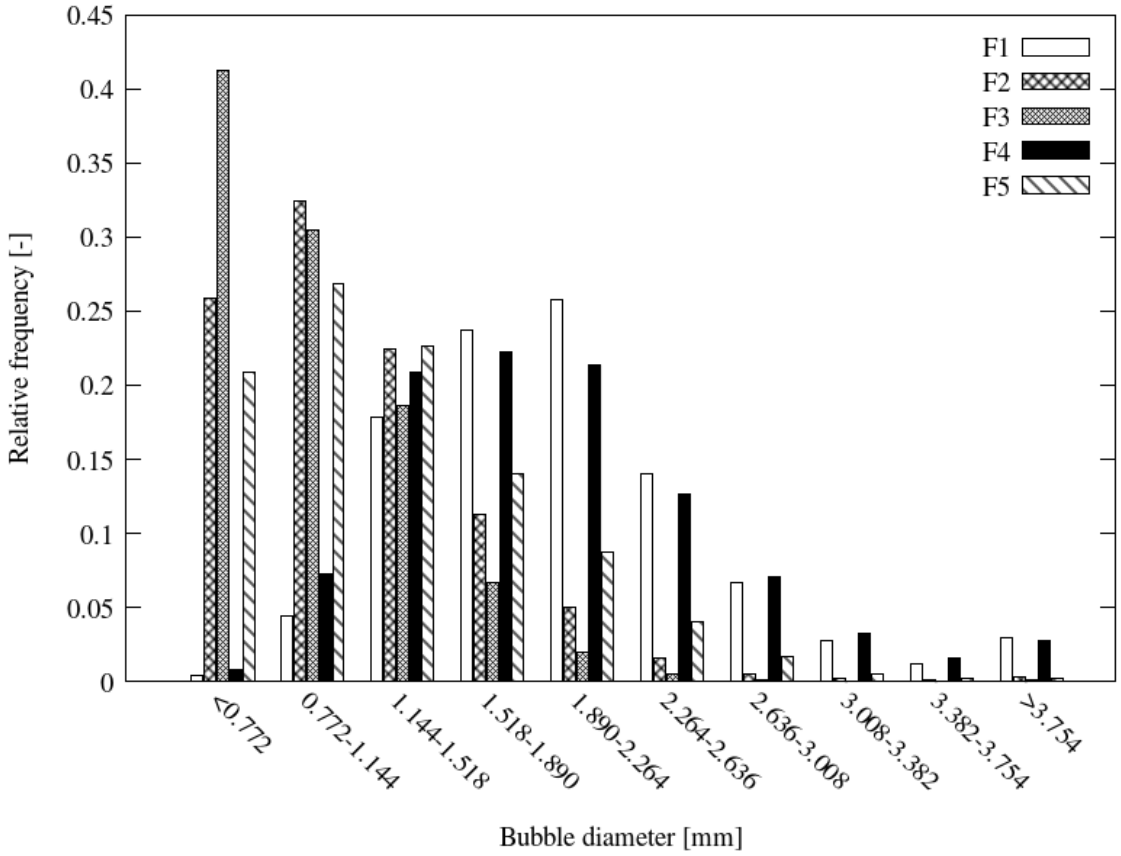

Figure 16: Bubble size distribution for the 5 cases (number density).

The flow regime in the 40 m³ reactor according to the flow map in Figure 12 should be flooded although the simulation did not show a flooded behavior. Holdup measurements will be conducted to validate the correctness of the flow regime prediction. As expected the simulation wall time was significantly higher. The first 50,000 time steps took 19.7 h compared to one hour for the 150 l reactor simulations on average.

After 190,000 time steps or 15.8 s in real time the reactor contained 4.48 million bubble parcels which corresponds to a gas holdup of 5.87 %. The startup of the reactor can be seen in Figure 17.

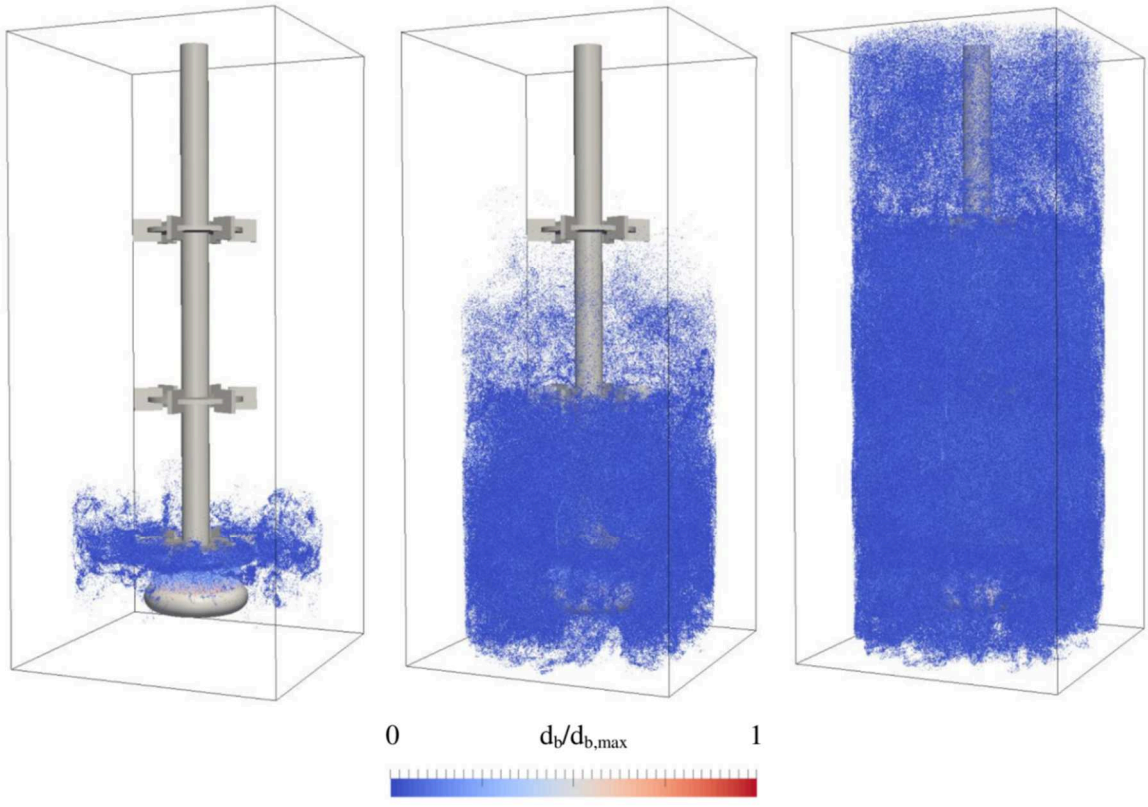

Figure 17: Bubble distribution after 1.66, 6.23 and 15.76 s in a 40m³ reactor

In the large scale reactor the effect of compartment filling is more pronounced than in the small reactor. The recirculation zones around each stirrer are filled with bubbles before the majority of the bubbles rise to the next stirrer. This can be clearly seen in the snapshots in Figure 17.

3.5 Conclusions

A modeling approach for a stirred and aerated reactor was developed and is described in detail in this publication. A D3Q19 lattice Boltzmann method (LBM) for the fluid flow field calculations and the Lagrangian particle tracking (LPT) for the bubble movement were employed. A large eddy simulation was implemented to simulate turbulence via a Smagorinsky subgrid model. The modified bounce-back method was used to integrate the static and rotating boundaries. Bubble coalescence and breakup were included in the model. These localized algorithms take advantage of the calculation power of graphics processors (GPU) with the CUDA technology and C++. A parcel approach was applied to group bubbles with similar properties together. To avoid the computationally costly search for direct collision partners, we applied a stochastic approach to determine the rate and impact point of a bubble-bubble collision based on the statistics around the location of the bubble parcel. The occurrence of breakup and coalescence was based on probabilities and, in the latter case, required a fictional collision partner. For the first time a detailed and validated simulation of an aerated industrial sized reactor is possible in a timeframe suitable for reactor design and engineering.

To obtain data for the validation of the model, a 150 l lab-scale stirred and aerated acrylic glass reactor with three Rushton turbines was constructed. To measure the local holdup distribution, a needle-shaped conductivity sensor was used. The holdup values of five operating points of the reactor were measured. The simulation data was in good agreement with the experimental data, especially in the operating points with a low aeration rate and a low stirrer speed.

With the validated simulation program bubble size distributions of the flow regimes were extracted and analyzed. The comparison showed a strong influence of the stirrer

rotational speed on the size distribution peak. As expected, the faster the stirrer turns, the lower is the average bubble diameter. The gas flow rate had almost no impact on the distribution. The k_La was calculated with the Higbie surface renewal theory. As a well-mixed solution is assumed in the experiment the k_La values in the simulation are higher than the experimental ones at the high holdup cases.

The simulation of a 40m³ reactor with three Rushton turbines showed an almost twentyfold increase in simulation time. A detailed comparison with industrial size reactor holdups is subject to further research.

To further improve the modelling of the k_La value a species transport solver will be implemented. The dissolved oxygen in the fluid is then included in the calculation of the oxygen mass transfer from the bubble to the fluid. Predictions of the shear rate and the power consumption are also subjects to further research.

3.6 Nomenclature

A_b – bubble surface, m^2

A_{fr} - frontal area of the bubble, m^2

C – stirrer height from bottom, m

c_D – drag force coefficient

c_L – lift force coefficient

C_l – turbulence correction constant

C_{SM} – Smagorinsky constant

d_b – bubble diameter, m

D – stirrer diameter, m

D_A – mass diffusivity, m^2/s

Eo – Eötvös number

$\vec{e}_a$ – spatial direction

ε – turbulent energy dissipation, m^2/s^3

f_a^{eq} – equilibrium function

f_a – statistical distribution function

$f_{a'}$ – statistical distribution function after the bounce-back

F – force, N

$\vec{g}$ – gravitational acceleration, m/s^2

H – reactor height, m

μ – viscosity, Pa·s

m_f – mass of the fluid of the same volume as the bubble, kg

m_g – bubble mass, kg

n – influence radius (mapping function, $1.5\,d_b$), m

Q – momentum flux

ρ_b – air density, kg/m³

ρ – fluid density, kg/m³

r^* – dimensionless radius

Re – Reynolds number

s – gas sparger height, m

Sc – Schmidt number

Sh – Sherwood number

σ – surface tension, N/m

t^* – dimensionless time

T – reactor diameter, m

τ – relaxation factor

t – time, s

$\vec{u}$ – fluid velocity, m/s

$\vec{u}_b$ – solid part velocity, m/s

$\vec{v}$ – bubble velocity, m/s

w_a – specific weighting factor

$\vec{x}$ – fluid node position (mapping function), m

$\vec{x}_p$ – bubble position (mapping function), m

3.7 References

Aghbolaghy M, Karimi A. 2014. Simulation and optimization of enzymatic hydrogen peroxide production in a continuous stirred tank reactor using CFD-RSM combined method. *J. Taiwan Inst. Chem. Eng.* **45**:101–107.

Akira M. 1986. A mixed finite element method for boundary flux computation. . Article *Comput. Methods Appl. Mech. Eng.* **57**:239–243.

Alves SS, Maia CI, Vasconcelos JMT. 2004. Gas-liquid mass transfer coefficient in stirred tanks interpreted through bubble contamination kinetics. *Chem. Eng. Process. Process Intensif.* **43**:823–830.

Arlov D, Revstedt J, Fuchs L. 2008. Numerical simulation of a gas–liquid Rushton stirred reactor–LES and LPT. *Comput. Fluids* **37**:793–801.

Bakker A, Oshinowo LM. 2004. Modelling of turbulence in stirred vessels using large eddy simulation. *Chem. Eng. Res. Des.* **82**:1169–1178.

Bartels C, Breuer M, Durst F. 2000. Comparison between Direct Numerical Simulation and ke Prediction of the Flow in a Vessel Stirred by a Rushton Turbine. In: *10th Eur. Conf. Mix. Proc.*, pp. 239–243.

Bashiri H, Heniche M, Bertrand F, Chaouki J. 2013. Compartmental modelling of turbulent fluid flow for the scale-up of stirred tanks. *Can. J. Chem. Eng.* **92**:1070–1081.

Besbes S, El Hajem M, Ben Aissia H, Champagne JY, Jay J. 2015. PIV measurements and Eulerian–Lagrangian simulations of the unsteady gas–liquid flow in a needle sparger rectangular bubble column. *Chem. Eng. Sci.* **126**:560–572.

Bhatnagar PL, Gross EP, Krook M. 1954. A model for collision process in gases. I. Small amplitude process in charged and neutral one-component system. *Phys. Rev.* **94**:511.

Bombač A, Žun I. 1997. Gas-filled cavity structures and local void fraction distribution in aerated stirred vessel. *AIChE J.* **43**:2921–2931.

Bombač A, Žun I. 2006. Individual impeller flooding in aerated vessel stirred by multiple-Rushton impellers. *Chem. Eng. J.* **116**:85–95.

Bouzidi M, Firdaouss M, Lallemand P. 2001. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Phys. Fluids* **13**:3452.

Brucato A, Ciofalo M, Grisafi F, Micale G. 1998. Numerical prediction of flow fields in baffled stirred vessels: A comparison of alternative modelling approaches. *Chem. Eng. Sci.* **53**:3653–3684.

Buick JM. 2009. Lattice Boltzmann simulation of power-law fluid flow in the mixing section of a single-screw extruder. *Chem. Eng. Sci.* **64**:52–58.

Chen S, Doolen GD. 1998. Lattice Boltzmann Method for Fluid Flows. *Annu. Rev. Fluid Mech.* **30**:329–364.

Chu L, Robinson DK. 2001. Industrial choices for protein production by large-scale cell culture. *Curr. Opin. Biotechnol.* **12**:180–187.

Darmana D, Deen NG, Kuipers J a. M. 2006. Detailed 3D Modeling of Mass Transfer Processes in Two-Phase Flows with Dynamic Interfaces. *Chem. Eng. Technol.* **29**:1027–1033.

Deen NG, Solberg T, Hjertager BH. 2002. Flow generated by an aerated Rushton

impeller: two-phase PIV experiments and numerical simulations. *Can. J. Chem. Eng.* **80**:1–15.

Deglon D a., Meyer CJ. 2006. CFD modelling of stirred tanks: Numerical considerations. *Miner. Eng.* **19**:1059–1068.

Derksen JJ. 2003. Numerical simulation of solids suspension in a stirred tank. *AIChE J.* **49**:2700–2714.

Derksen JJ. 2012. Highly resolved simulations of solids suspension in a small mixing tank. . Article *AIChE J.* **58**:3266–3278.

Derksen JJ, van den Akker HEA. 1999. Large eddy simulations on the flow driven by a Rushton turbine. *AIChE J.* **45**:209–221.

Derksen JJ, Kooman JL, van den Akker HEA. 1997. Parallel fluid flow simulations by means of a lattice-Boltzmann scheme. In: *High-Performance Comput. Netw.*, pp. 524–530.

Dhotre MT, Deen NG, Niceno B, Khan Z, Joshi JB. 2013. Large eddy simulation for dispersed bubbly flows: A review. *Int. J. Chem. Eng.* **2013**:1–22.

Eggels JGM, Somers J a. 1995. Numerical simulation of free convective flow using the lattice-Boltzmann scheme. *Int. J. Heat Fluid Flow* **16**:357–364.

Eggels J. 1996. Direct and large-eddy simulation of turbulent fluid flow using the lattice-Boltzmann scheme. *Int. J. heat fluid flow* **17**:307–323.

Eng M, Rasmuson A. 2012. Influence of solids on macro-instabilities in a stirred tank. *Chem. Eng. Res. Des.* **90**:1052–1062.

Feng ZG, Michaelides EE. 2004. The immersed boundary-lattice Boltzmann method for

solving fluid-particles interaction problems. *J. Comput. Phys.* **195**:602–628.

Germano M, Piomelli U, Moin P, Cabot WH. 1991. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids A* **3**:1760–1765.

Gillissen JJJ, van den Akker HEA. 2009. Direct Numerical Simulation of a Turbulently Stirred Tank. . Inproceedings In: *13th Eur. Conf. Mix. London*, pp. 14–17.

Gillissen JJJ, Van den Akker HEA. 2012. Direct numerical simulation of the turbulent flow in a baffled tank driven by a Rushton turbine. *AIChE J.* **58**:3878–3890.

Guo Z, Zheng C, Shi B. 2002a. An extrapolation method for boundary conditions in lattice Boltzmann method. *Phys. Fluids* **14**:2007–2010.

Guo Z, Zheng C, Shi B. 2002b. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **65**:1–6.

Han L, Cao Y, Wu X, Bai G, Liu Y. 2010. The Effect of Eddy-bubble Interaction Model on the Turbulent Dispersion of Gas Bubbles in Stirred Tanks. *Chinese J. Chem. Eng.* **18**:27–33.

Han L, Liu Y, Luo H. 2007. Numerical simulation of gas holdup distribution in a standard Rushton stirred tank using discrete particle method. *Chinese J. Chem. Eng.* **15**:808–813.

Harlow FH, Welch JE. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids* **8**:2182–2189.

Hartmann H, Derksen JJ, van den Akker HE a. 2004a. Macroinstability uncovered in a Rushton turbine stirred tank by means of LES. *AIChE J.* **50**:2383–2393.

Hartmann H, Derksen JJ, Montavon C, Pearson J, Hamill IS, van den Akker HEA.

2004b. Assessment of large eddy and RANS stirred tank simulations by means of LDA. *Chem. Eng. Sci.* **59**:2419–2432.

Harvey PS, Greaves MG. 1982a. Turbulent Flow in an Agitated Vessel Part-II: Numerical Solution and Model Predictions. *Chem. Eng. Res. Des.* **60**:201–210.

Harvey PS, Greaves MG. 1982b. Turbulent Flow in an Agitated Vessel Part-I: A Predictive Model. *Chem. Eng. Res. Des.* **60**:195–200.

Higbie R. 1935. The role of a pure gas into a still liquid during short priod of exposure. . Book *Transm.Am.Inst.Chem.Engr.* **35**:365–373.

Hu G, Celik I. 2008. Eulerian–Lagrangian based large-eddy simulation of a partially aerated flat bubble column. *Chem. Eng. Sci.* **63**:253–271.

Hutmacher DW, Singh H. 2008. Computational fluid dynamics for improved bioreactor design and 3D culture. *Trends Biotechnol.* **26**:166–172.

Jordan PM. 1995. Bioprocess engineering principles. . London: Academic Press Ltd.

Joshi JB, Nere NK, Rane C V., Murthy BN, Mathpati CS, Patwardhan AW, Ranade V V. 2011. CFD simulation of stirred tanks: Comparison of turbulence models. Part I: Radial flow impellers. *Can. J. Chem. Eng.* **89**:23–82.

Koynov A, Tryggvason G, Khinast JG. 2007. Characterization of the Localized Hydrodynamic Shear Forces and Dissolved Oxygen Distribution in Sparged Bioreactors. *Biotechnol. Bioeng.* **97**:317–331.

Kresta SM, Wood PE. 1991. Prediction of the three-dimensional turbulent flow in stirred tanks. *AIChE J.* **37**:448–460.

Krishna R, Van Baten JM, Urseanu MI. 2000. Three-phase Eulerian simulations of

bubble column reactors operating in the churn-turbulent regime: A scale up strategy. *Chem. Eng. Sci.* **55**:3275–3286.

Ladd A. 1994. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *J. Fluid Mech.* **211**:285–309.

Lapin A, Lübbert A. 1994. Numerical simulation of the dynamics of two-phase gas—liquid flows in bubble columns. *Chem. Eng. Sci.* **49**:3661–3674.

Lau YM, Bai W, Deen NG, Kuipers J a M. 2014. Numerical study of bubble break-up in bubbly flows using a deterministic Euler-Lagrange framework. *Chem. Eng. Sci.* **108**:9–22.

Li Z, Derksen JJ, Gao Z. 2015. Models and Applications for Simulating Turbulent Solid&Liquid Suspensions in Stirred Tanks. *J. Chem. Eng. Japan* **48**:329–336.

Liew E, Nandong J, Samyudia Y. 2008. Unification of Mixing and Cellular Metabolism in Bioreactor Modeling: A Multi-scale Approach. In: *RSCE-SOMCHE*. Kuala Lumpur, Malaysia, pp. 9–16.

Luo H, Svendsen HF. 1996. Theoretical Model for Drop and Bubble Breakup in Turbulent Dispersions. *AIChE J.* **42**:1225–1233.

Mathpati CS, Joshi JB. 2007. Insight into theories of heat and mass transfer at the solid-fluid interface using direct numerical simulation and large eddy simulation. In: *Ind. Eng. Chem. Res.*, Vol. 46, pp. 8525–8557.

Mei R, Luo L-S, Shyy W. 1999. An Accurate Curved Boundary Treatment in the Lattice Boltzmann Method. *J. Comput. Phys.* **155**:307–330.

Morchain J, Gabelle J-C, Cockx A. 2014. A coupled population balance model and

CFD approach for the simulation of mixing issues in lab-scale and industrial bioreactors. *AIChE J.* **60**:27–40.

Ng K, Fentiman NJ, Lee KC, Yianneskis M. 1998. Assessment of Sliding Mesh CFD Predictions and LDA Measurements of the Flow in a Tank Stirred by a Rushton Impeller. *Chem. Eng. Res. Des.* **76**:737–747.

Nikolai TJ, Hu WS. 1992. Cultivation of mammalian cells on macroporous microcarriers. *Enzyme Microb. Technol.* **14**:203–208.

Obrecht C, Kuznik F, Tourancheau B, Roux J-J. 2011. A new approach to the lattice Boltzmann method for graphics processing units. *Comput. Math. with Appl.* **61**:3628–3638.

Obrecht C, Kuznik F, Tourancheau B, Roux J-J. 2013. Multi-GPU implementation of the lattice Boltzmann method. *Comput. Math. with Appl.* **65**:252–261.

Patankar S. 1980. Numerical heat transfer and fluid flow. . Book. CRC Press.

Pericleous KA, Patel MK. 1987. The Modelling of Tangential and Axial Agitators in Chemical Reactors. *Physicochem. Hydrodyn.* **8**:105–123.

Perng CY, Murthy JY. 1993. A moving-deforming-mesh technique for simulation of flow in mixing tanks. *AIChE Symp. Ser.* **89**:37.

Petitti M, Vanni M, Marchisio DL, Buffo A, Podenzani F. 2013. Simulation of coalescence, break-up and mass transfer in a gas-liquid stirred tank with CQMOM. *Chem. Eng. J.* **228**:1182–1194.

Qian Y, D'Humières D, Lallemand P. 1992. Lattice BGK models for Navier-Stokes equation. *EPL (Europhysics Lett.)* **17**:479–484.

Radl S, Khinast JG. 2007. Prediction of mass transfer coefficients in non-Newtonian fermentation media using first-principles methods. *Biotechnol. Bioeng.* **97**:1329–34.

Radl S, Khinast JG. 2010. Multiphase flow and mixing in dilute bubble swarms. . Article *AIChE J.* **56**:2421–2445.

Ranade VV, Perrard M, Le Sauze N, Xuereb C, Bertrand J. 2001. Trailing Vortices of Rushton Turbine. *Chem. Eng. Res. Des.* **79**:3–12.

Ranade V V, Joshi JB. 1990. Flow generated by a disc turbine. II: Mathematical modelling and comparison with experimental data. . Article *Chem. Eng. Res. Des.* **68**:34–50.

Rani KY, Rao VSR. 1999. Control of fermenters - A review. *Bioprocess Eng.* **21**:77–88.

Richardson LF. 1911. The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **210**:307–357.

Roubos J a., Van Straten G, Van Boxtel a. JB. 1999. An evolutionary strategy for fed-batch bioreactor optimization; concepts and performance. *J. Biotechnol.* **67**:173–187.

Roy S, Acharya S. 2012. Scalar mixing in a turbulent stirred tank with pitched blade turbine: Role of impeller speed perturbation. *Chem. Eng. Res. Des.* **90**:884–898.

Sbrizzai F, Lavezzo V, Verzicco R, Campolo M, Soldati A. 2006. Direct numerical simulation of turbulent particle dispersion in an unbaffled stirred-tank reactor.

Chem. Eng. Sci. **61**:2843–2851.

Smagorinsky J. 1963. General circulation experiments with the primitive equations: I. The basic experiment. *Mon. Weather Rev.* **91**:99–164.

Sokolichin A, Eigenberger G. 1994. Gas—liquid flow in bubble columns and loop reactors: Part I. Detailed modelling and numerical simulation. *Chem. Eng. Sci.* **49**:5735–5746.

Sommerfeld M. 2003. Euler/Lagrange calculations of bubbly flows with consideration of bubble coalescence. *Can. J. Chem. Eng.* **81**:508–518.

Stobiac V, Tanguy P a., Bertrand F. 2013. Boundary conditions for the lattice Boltzmann method in the case of viscous mixing flows. *Comput. Fluids* **73**:145–161.

Stobiac V, Tanguy P a., Bertrand F. 2014. Investigation of the accuracy of the extrapolation method for the lattice Boltzmann simulation of viscous fluid flow in a Maxblend impeller system. *Comput. Chem. Eng.* **60**:112–123.

Sukop M, Thorne D. 2006. Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers. . Berlin, Heidelberg: Springer.

Sungkorn R, Derksen JJ, Khinast JG. 2011. Modeling of turbulent gas–liquid bubbly flows using stochastic Lagrangian model and lattice-Boltzmann scheme. *Chem. Eng. Sci.* **66**:2745–2757.

Sungkorn R, Derksen JJ, Khinast JG. 2012a. Euler–Lagrange modeling of a gas–liquid stirred reactor with consideration of bubble breakage and coalescence. *AIChE J.* **58**:1356–1370.

Sungkorn R, Derksen JJ, Khinast JG. 2012b. Modeling of aerated stirred tanks with shear-thinning power law liquids. *Int. J. Heat Fluid Flow* **36**:153–166.

Tölke J. 2008. Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA. *Comput. Vis. Sci.* **13**:29–39.

Tomiyama A. 1998. Struggle with computational bubble dynamics. *Multiph. Sci. Technol.* **10**:369–405.

Torvik R, Svendsen HF. 1990. Modelling of slurry reactors. A fundamental approach. *Chem. Eng. Sci.* **45**:2325–2332.

Treffer D. 2011. Development of a measurement technique to investigate mixing behavior and gas hold-up distribution in a multiphase reactor. .

Verzicco R, Fatica M, Iaccarino G, Orlandi P. 2004. Flow in an impeller-stirred tank using an immersed-boundary method. *AIChE J.* **50**:1109–1118.

Wang H, Jia X, Wang X, Zhou Z, Wen J, Zhang J. 2014. CFD modeling of hydrodynamic characteristics of a gas–liquid two-phase stirred tank. *Appl. Math. Model.* **38**:63–92.

Warnock JN, Al-Rubeai M. 2006. Bioreactor systems for the production of biopharmaceuticals from animal cells. *Biotechnol. Appl. Biochem.* **45**:1–12.

Webb C, Que F, Senior PR. 1992. Dynamic simulation of gas-liquid dispersion behaviour in a 2-D bubble column using a graphics mini-supercomputer. *Chem. Eng. Sci.* **47**:3305–3312.

Wolf-Gladrow DA. 2000. Lattice-gas cellular automata and lattice Boltzmann models: An Introduction. . Book. Springer Science & Business Media.

Wu J, Shu C. 2009. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *J. Comput. Phys.* **228**:1963–1979.

Yeoh SL, Papadakis G, Yianneskis M. 2005. Determination of mixing time and degree of homogeneity in stirred vessels with large eddy simulation. *Chem. Eng. Sci.* **60**:2293–2302.

Yeoh SL, Papadakis G, Yianneskis M. 2004. Numerical Simulation of Turbulent Flow Characteristics in a Stirred Vessel Using the LES and RANS Approaches with the Sliding/Deforming Mesh Methodology. *Chem. Eng. Res. Des.* **82**:834–848.

Yu H, Girimaji SS, Luo L-S. 2005. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *J. Comput. Phys.* **209**:599–616.

Zhang Q, Yang C, Mao Z-S, Mu J. 2012. Large Eddy Simulation of Turbulent Flow and Mixing Time in a Gas–Liquid Stirred Tank. *Ind. Eng. Chem. Res.* **51**:10124–10131.

4

Species Transport in Industrial-scale Aerated Bioreactors

Although large-scale aerated stirred bioreactors are used in many industrial sectors, their design and scale-up is largely based on empirical knowledge. Simulations via Computational Fluid Dynamics (CFD) on graphical processing units (GPUs) provide a better understanding of the inner workings of the reactors. The fluid and bubble dynamics can serve as a basis for simulations of biological processes, including the conversion of substrate and dissolved oxygen to carbon dioxide by microorganisms. To transport the quantities required for such a simulation, a species transport solver can be used. For achieving the highest efficiency on the GPUs, the lattice Boltzmann method can be applied to discretize the transport equation. Obtaining the Bhatnagar–Gross–Krook equilibrium distribution functions via the Chapman Enskog expansion of the transport equation may result in negative concentration values and a mass gain due to fluid acceleration. The present thesis introduces a novel way of using the lattice Boltzmann method as a basis for the species transport simulation, which eliminates the above-mentioned shortcomings. The new method was tested via a Gaussian peak transport and applied to transporting nutrients in the reactor. Single feed versus split feed of substrate, the dissolution and the transport of dissolved oxygen were examined. The simulation speed and the memory consumption were compared to particle-based transport solvers. A simple biological model was implemented to demonstrate the ability of the code to simulate biological processes inside a 150 l reactor.

4.1 Introduction

In order to perform simulations of industrial-size bioreactors that are fast enough to be used for the purposes of the reactor's design process, every part of the simulation algorithm has to be carefully selected and evaluated. To that end, graphic cards and graphic processing units (GPUs) are promising since the number of calculation cores on a single GPU is three to four orders of magnitude higher than those on a central processing unit (CPU), which is typically used for simulations. Moreover, all the calculation cores on the GPU have access to the same memory, eliminating a communication step of broadcasting the calculation results from various processing units to other processors participating in the simulation (e.g., in a scientific cluster). This communication step limits the achievable simulation speedup when more and more CPU cores are employed (i.e., parallelization) to reduce computational effort. In contrast, with the Compute Unified Device Architecture (CUDA) technology and a large number of calculation cores on GPUs with their shared memory efficient large-scale simulations can be carried out.

Nevertheless, one peculiarity of GPU has to be taken into account: the algorithm must have a high grade of localization. The less information from other nodes the algorithm requires, the faster it can be executed. This is why the lattice Boltzmann method (LBM) is typically chosen for the fluid field calculation (Bailey et al., 2009; Chen and Doolen, 1998b; Obrecht et al., 2011; Obrecht et al., 2013a; Obrecht et al., 2013b; Sungkorn et al., 2011; Sungkorn et al., 2012a; Sungkorn et al., 2012b; Tölke, 2008; Witz et al., 2016; Witz and Khinast, 2015) and the Lagrangian particle tracking is used to simulate the bubble movement in the reactor (Sungkorn et al., 2011; Sungkorn et al., 2012a;

Sungkorn et al., 2012b; Witz et al., 2016; Witz and Khinast, 2015). In addition to the hydrodynamics, mass transport in the bioreactor is of paramount importance for controlling and optimizing the performance and for enabling rational scale-up.

One way to simulate the species transport in bioreactors is assigning a certain quantity of species (solute) to a massless tracer particle (Devkota and Imberger, 2009; Ponoth and Mclaughlin, 2000). Then, a Lagrangian method can be used to calculate the flow-field-guided movement of those tracer particles. The problem with this method is that the resolution of the particle-based species transport solver is directly related to the number of particles, and thus, the memory consumption. The more particles are used, the less solute is represented by one particle. Moreover, when simulating a steady inflow of substrate, the memory consumption increases with the calculation time, which is problematic since at some point the available memory is exhausted. In contrast, a lattice-Boltzmann-based solver could eliminate this problem since the resolution is determined by the datatype used for storing the species number at the nodes. Because the node number is constant throughout the simulation, the memory consumption is constant as well. Moreover, the algorithm would be as localized as the algorithm for solving the continuous-phase hydrodynamics.

The use of a lattice-Boltzmann method for solving the species transport equation has extensively been reported in the literature (Bin et al., 2005; Camas, 2008; Chai and Zhao, 2013; Chai and Zhao, 2014; Chopard et al., 2009; Eggels and Somers, 1995; Gebäck and Heintz, 2013; Ginzburg, 2005a; Ginzburg, 2005b; Guo et al., 2002a; Perko and Patel, 2014; Ponce Dawson et al., 1993; Rasin et al., 2005; Riaud et al., 2014; Shi et al., 2008; van der Sman and Ernst, 2000; Yoshida and Nagaoka, 2010; Zhang et al., 2002; Zhang et al., 2011). In the majority of these publications, the Bhatnagar–Gross–

Krook (BGK) equilibrium distribution functions (Bhatnagar et al., 1954), which are common in fluid field calculations, are utilized. Chai and Zhao (Chai and Zhao, 2013) reported a derivation of the species transport equation and the Chapman Enskog expansion. However, this approach leads to problems as negative concentration values are observed (Karimi and Nakshatrala, 2016), as well as mass gain when the fluid is accelerated by a pressure gradient (Chai and Zhao, 2013). Several efforts to correct this problem that have been made (Chai and Zhao, 2013; Chai and Zhao, 2014; Chopard et al., 2009; Ginzburg, 2005a; Ginzburg, 2005b).

In this work, an algorithm is introduced to solve the transport equations to circumvent these problems while maintaining compatibility with the existing lattice Boltzmann fluid node grid and preserving the highly parallelizable and localized properties of the lattice Boltzmann method. It should be noted that our method still requires more memory than traditional ones (although less than particle-based methods) and that the numerical diffusion is not as low as in established methods like the total variation diminishing scheme (Harten, 1997). The new method should serve as a basis for implementing diffusion models and numerical diffusion limiting algorithms.

The reactor setup, the fluid field and the bubble movement simulation algorithms are detailed in Section 4.2.1. The new species transport algorithm is presented in Section 4.2.2. In Section 4.3, the memory consumption and the calculation wall time of the species transport solver are compared with those of a particle-based solver. Additionally, a Gaussian hill transport was conducted to measure the numerical diffusion and a 150l gassed bioreactor equipped with three Rushton turbines were chosen to test the transient substrate and oxygen distribution. Single feed and split feed

were compared for two different injection sites. Finally, results are presented for a simple biological model.

4.2 Modeling

4.2.1 Multi-phase reactor simulation

The fluid field was simulated using a D3Q19 lattice Boltzmann method (LBM) with 3 dimensions and 19 density distribution functions, including the non-moving resting function. The algorithm consists of the collision step with the distribution functions relaxed towards the equilibrium distribution function, and the streaming step with the distribution functions transported to the neighboring cell in their spatial direction. The equilibrium distribution function is calculated using the BGK (Bhatnagar-Gross-Krook) approximation (Qian et al., 1992). Forces acting on the fluid, including the backward coupling force of the bubbles, are added using the formulation proposed by Guo (Guo et al., 2002c). The boundaries are modelled via the modified bounce-back method (Ladd, 1994). Since the flow field is highly turbulent, a large eddy simulation together with the Smagorinsky subgrid method are applied to take the effects of turbulence into account (Smagorinsky, 1963; Yu et al., 2005).

To calculate the bubble movement, the sum of the fluid forces including gravity, drag, lift and added mass forces were computed (Hu and Celik, 2008). For bubble break-up and coalescence a statistical approach was used, grouping bubbles with similar properties into parcels (Sommerfeld, 2003). A fictitious bubble is created based on the statistics in the neighborhood of the bubble parcel. This fictitious bubble is the collision or coalescence partner of the bubble that represented the parcel. Turbulent eddies in the size range of the bubble diameter were considered the main source of bubble break-up.

The model of Luo and Svendsen, 1996 was applied to determine the daughter bubble size distribution.

To test the multi-phase simulation and the species transport solver, a 150l reactor with a ring sparger, four baffles and three Rushton turbines was used. More information about the simulation algorithms and the validation experiments in a 150l reactor can be found in (Witz et al., 2016).

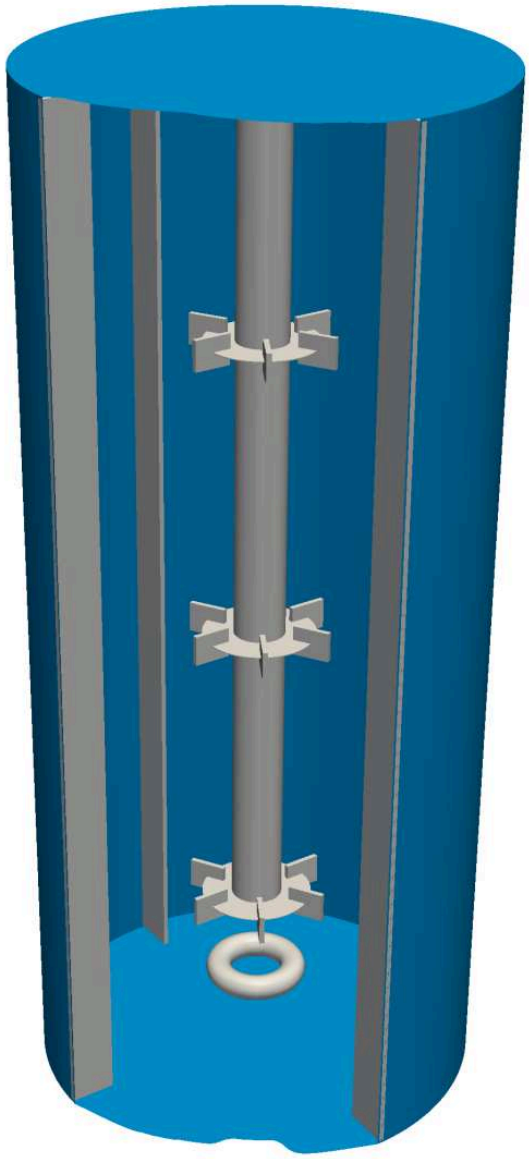

Figure 18: Reactor setup with three Rushton turbines, four baffles and a ring sparger

Table 10: Simulation parameters

Reactor height	H	1015 mm
Reactor diameter	T	440 mm
Distance between stirrers	h	290 mm
Bottom clearance of lowest stirrer	c	145 mm
Stirrer outer diameter	D	147 mm
Disk diameter		110 mm
Stirrer blade height		30 mm
Stirrer blade width		37 mm
Shaft diameter		50 mm
Air inlet height	s	73 mm
Baffles number		4
Gas flow rate		0.2 vvm
Stirrer speed		190 1/min
Headspace pressure		101.325 kPa
Mole fraction oxygen in gas bubbles	x_{O_2}	0.21
Diffusion coefficient oxygen in water	D_{A,O_2}	$2.1 \cdot 10^{-9}$ m²/s
Diffusion coefficient of carbon dioxide in water	D_{A,CO_2}	$1.92 \cdot 10^{-9}$ m²/s
Henry constant for oxygen in water	K_{H,O_2}^{px}	42795.18 Pa
Concentration of water	c_w	55.56 kmol/m³

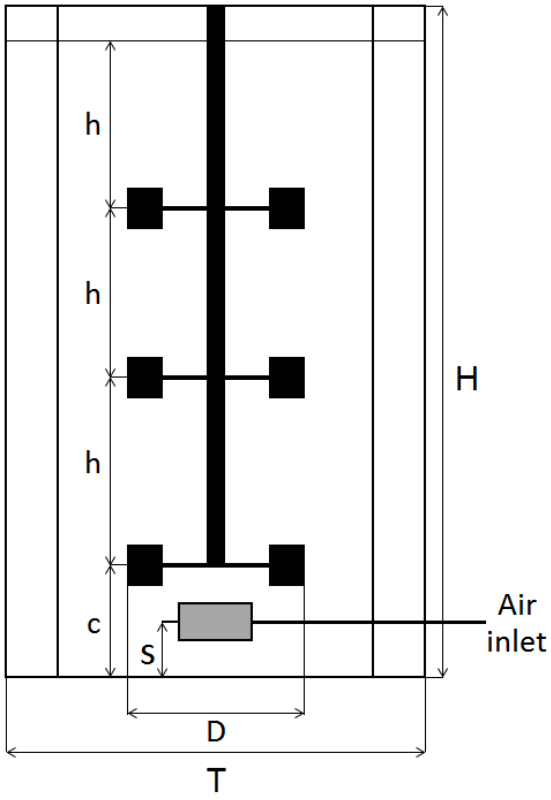

Figure 19: Reactor geometry

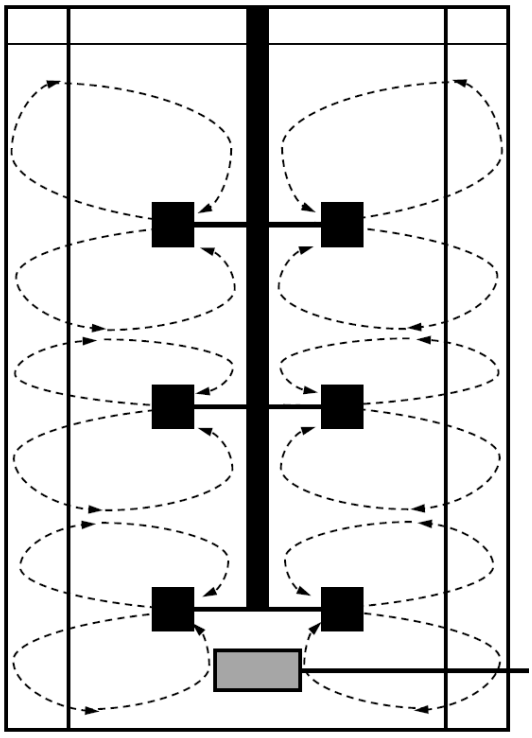

Figure 20: Sketch of the fluid flow vortex structure in the reactor.

4.2.2 Species transport using the lattice Boltzmann method

The cause of the well-known problems using the lattice Boltzmann method for species transport like negative concentration values (Karimi and Nakshatrala, 2016) and mass gain in accelerated fluids (Chai and Zhao, 2013) was investigated by comparing the fluid field with the species transport calculation:

In the classical D3Q19 LBM the values of the statistical distribution functions (f_a) are calculated using the following formulas:

$$f_a\left(\vec{x}+\vec{e}_a\Delta t, t+\Delta t\right)-f_a\left(\vec{x},t\right)=-\frac{f_a\left(\vec{x},t\right)-f_a^{eq}\left(\vec{x},t\right)}{\tau}+\Delta t F_a\left(\vec{x},t\right) \tag{37}$$

$$f_a^{eq}\left(\vec{x},t\right)=w_a\rho\left[1+\frac{\vec{e}_a\cdot\vec{u}}{c_s^2}+\frac{\left(\vec{e}_a\cdot\vec{u}\right)^2}{2c_s^4}-\frac{\vec{u}^2}{2c_s^2}\right] \tag{38}$$

$$\tau=\frac{\nu}{\Delta t c_s^2}+\frac{1}{2} \tag{39}$$

$$\vec{e}_a=\begin{cases}(0,0,0) & \text{a=0}\\ (\pm1,0,0),(0,\pm1,0),(0,0,\pm1) & \text{a=1-6}\\ (\pm1,\pm1,0),(\pm1,0,\pm1),(0,\pm1,\pm1) & \text{a=7-18}\end{cases} \tag{40}$$

Where a denotes the different spatial directions, $\vec{e}_a$ are the unit vectors, w_a are the weighting factors and f_a^{eq} stands for the equilibrium distribution functions. τ is the relaxation factor, which is calculated using the fluid's viscosity ν. F_a accounts for the external forces or, in the case of the species transport, includes the sources or sinks of species mass. $c_s=\left(1/\sqrt{3}\right)\Delta x/\Delta t$ denotes the speed of sound. Δx is the lattice spacing and Δt is the time step.

When using the lattice Boltzmann method, the fluid field is simulated by transporting the mass, which is represented by the fluid's density ρ, via the lattice Boltzmann scheme (Eq. (37)) (Bhatnagar et al., 1954; Chen and Doolen, 1998c).

Now for the advective species transport simulation using the lattice Boltzmann approach the same principle is applied, but instead of the fluid density, the species mass is transported, which is represented by the concentration of the solute (C instead of ρ).

To identify possible reasons for the aforementioned problems reported by several authors applying this algorithm, one could think of a setup of a resting fluid where a certain amount of species mass m_s is added to one cell via the source term. All other cells contain no species mass, initially. Then the following equation is used to calculate the new distribution functions:

$$f_a\left(\vec{x}+\vec{e}_a\Delta t, t+\Delta t\right)-f_a\left(\vec{x},t\right)=-\frac{f_a\left(\vec{x},t\right)-f_a^{eq}\left(\vec{x},t\right)}{\tau}+\Delta t w_a m_s\,. \tag{41}$$

As all distribution vectors f_a and equilibrium distribution vectors f_a^{eq} are zero initially, the resulting distribution vector is:

$$f_a\left(\vec{x}+\vec{e}_a\Delta t, t+\Delta t\right)=\Delta t w_a m_s\,. \tag{42}$$

This results in a fast diffusion-like distribution of a large fraction of the added species source mass (only the amount $w_0 m_s$ stays at the resting function on the node), contrarily to what is physically expected for the advective transport in resting fluids.

If we now neglect this problem with the fast diffusion of species source mass and assume, that the cell we are looking at contains the concentration $C = C_1$ and

throughout the whole domain the fluid velocity is $\vec{u} = \vec{u}_1$. Then the value of a vector pointing in the opposite direction of the fluid velocity is:

$$f_a\left(\vec{x}+\vec{e}_a\Delta t, t+\Delta t\right) = \frac{1}{\tau} w_a C_1 \left[1+\frac{\vec{e}_a\cdot\vec{u}_1}{c_s^2}+\frac{\left(\vec{e}_a\cdot\vec{u}_1\right)^2}{2c_s^4}-\frac{\vec{u}_1^{\,2}}{2c_s^2}\right] \tag{43}$$

This result is again contrary to the expected outcome, that no species mass should be transported against the fluid flow direction. The species transport is even negative if $\vec{e}_a$ and $\vec{u}_1$ are orthogonal because of the negative quadratic velocity term. When the quadratic terms for the equilibrium distribution function are neglected resulting in the popularly used form shown in Eq. (44), the transport against the fluid flow direction is still present.

$$f_a^{eq}\left(\vec{x},t\right) = w_a C\left[1+\frac{\vec{e}_a\cdot\vec{u}}{c_s^2}\right] \tag{44}$$

Moreover, the value of each distribution function after the collision step depends on the value of the distribution functions received from the neighboring cells. As such, the amount of species transported in a certain direction depends both on the macroscopic velocity and the quantity of species coming from the prior cell. Equation 45 illustrates the influence of the distribution vectors gathered from the neighboring cells in the last time step $f_a(\vec{x}, t)$ on the values of the distribution vectors for the next time step $f_a(\vec{x} + \vec{e}_a\Delta t, t + \Delta t)$ by rearranging equation 37:

$$f_a\left(\vec{x}+\vec{e}_a\Delta t, t+\Delta t\right) = \left(1-\frac{1}{\tau}\right) f_a\left(\vec{x},t\right) + \frac{1}{\tau} f_a^{eq}\left(\vec{x},t\right) \tag{45}$$

This result is unphysical because the species does not have any inertia and is only transported via the macroscopic fluid velocity, so the value of $f_a(\vec{x} + \vec{e}_a\Delta t, t + \Delta t)$ can only depend on the species mass C in the cell but not on $f_a(\vec{x}, t)$. However, the direction where the species mass is coming from must not influence the direction where the species mass is transported to. To tackle these problems of fast diffusion and virtual inertia, some authors introduced correction factors (Chai and Zhao, 2014), third order terms (Camas, 2008) or multiple relaxation time schemes (Ginzburg, 2005a).

We propose a different way to circumvent the problems mentioned above by getting rid of the equilibrium distribution. Instead, only the macroscopic fluid velocity determines which amount of the species mass is assigned to the distribution functions. As no quadratic terms are present, no species transport in any other direction than the one of the fluid flow will occur and no negative distribution vector values are created.

Instead of 19 density distribution functions, 26 distribution functions T were used. The direction vectors $\vec{e}_a$ for the probability functions (T) are:

$$\vec{e}_a = \begin{cases} (0,0,0) & a = 0 \\ (\pm 1,0,0), (0,\pm 1,0), (0,0,\pm 1) & a = 1-6 \\ (\pm 1,\pm 1,0), (\pm 1,0,\pm 1), (0,\pm 1,\pm 1) & a = 7-18 \\ (\pm 1,\pm 1,\pm 1) & a = 19-26 \end{cases} \tag{46}$$

In the beginning of the adapted collision step, the distribution functions are summed up to determine the number of species m in the cell. In this way, the information from which direction the parts of species mass came from is removed.

$$m = \sum_a T_a(\vec{x}, t) + m_s \tag{47}$$

The source and sink terms m_s are added to the species sum. This has the advantage that no fast diffusion-like distribution can occur, as the source mass is not added to the distribution vectors directly. Next, the species mass is allocated to the distribution functions based on the values of the macroscopic fluid velocity $\vec{u}_f$ in the current cell. Since the distance between cell centers and the time step length are normalized to those in the lattice unit system, the value of the fraction of the species mass that is transported in a certain direction is directly proportional to the corresponding fluid velocity.

In order to distribute the current species number to the density distribution functions, the following method was used: the plus/minus one component or components of the direction vector $\vec{e}_a$ determine the fraction of the total species number assigned to the distribution function. If two or more direction vectors are equal plus/minus one, the minimum of the respective velocities is used. The minimum value of velocities can be applied directly since the hypotenuse of velocities is divided by the distance between the respective nodes, resulting in the initial minimum value.

The distribution functions pointing to the edges of the cell ($\vec{e}_a = (\pm 1, \pm 1, \pm 1)$) are now ready for the streaming step. The other distribution functions have to be reduced by an amount $C_{a\alpha}$ to prevent assigning the fractions of species mass twice to functions which point in the same spatial direction.

$$C_{a\alpha} = \vec{e}_{a\alpha} \max_\beta \left(\min\left(\left|\vec{u}_{f\alpha}\right|, \left|\vec{u}_{f\beta}\right|\right)\right), \alpha, \beta = x, y, z;\ \alpha \neq \beta \text{ and } a = 1 - 6 \tag{48}$$

$$C_{a\alpha} = |\vec{e}_{a\alpha}| \min_\beta\left(\Theta_0\left(\vec{e}_{a\beta}\right)\max(0, \vec{u}_{f\beta}) + \Theta_0\left(-\vec{e}_{a\beta}\right)\left|\min(0, \vec{u}_{f\beta})\right| + \delta\left(\vec{e}_{a\beta}\right)\left|\vec{u}_{f\beta}\right|\right), \alpha, \beta = x, y, z \text{ and } a = 7 - 18 \tag{49}$$

$$C_{a\alpha} = 0, \alpha = x, y, z \text{ and } a = 18 - 26 \tag{50}$$

$$T_a(\vec{x}, t) = m \min_\alpha \big(\Theta_0(\vec{e}_{a\alpha}) \max(0, \vec{u}_{f\alpha}) + \Theta_0(-\vec{e}_{a\alpha}) \big| \min(0, \vec{u}_{f\alpha}) \big| + \delta(\vec{e}_{a\alpha}) N - C_{a\alpha} \big), \alpha = x, y, z \text{ and } a = 1 - 26 \tag{51}$$

$$T_0(\vec{x}, t) = m - \sum_a T_a(\vec{x}, t) \tag{52}$$

where Θ_0 is the Heavyside function with $\Theta_0(0) = 0$, δ is the Kronecker delta and N is a sufficiently high number to reliably exclude the terms of the minimum calculation where $\delta(\vec{e}_{a\alpha}) = 0$. For the boundary handling, the modified bounce-back method (MBB) (Ladd, 1994) is applied. The reflection of the distribution functions occurs halfway between the calculation nodes, resulting in a stair-step boundary. To render this effect negligible, the node number has to be high enough. If the fluid velocity field is calculated via the LBM, the boundary node assignments can be used directly. Additionally, since this boundary treatment preserves the locality of the species transport method, it is ideal for GPUs.

The proposed method simulates the advective transport. The diffusion can be safely ignored for high Peclet numbers, which compares the turbulent diffusion with the advection. In the case of our reactor the average Reynolds number is in the order of 10^4, the Schmidt number is in the order of 10^2 which results in a Peclet number in the order of 10^6. Hence, the diffusion can be ignored. The turbulent Peclet number can be estimated applying the turbulent viscosity in the order of 10^{-5} and the turbulent Schmidt number which is in the order of 1 (Hartmann et al., 2006) which results in a turbulent diffusion coefficient in the order of 10^{-5}. The Peclet number is then in the order of 10^3,

so also the turbulent diffusion can be estimated to have a negligible influence on the species transport for the conditions applied in the present work.

The mass transfer of oxygen into the liquid phase is calculated using Higbie‘s penetration theory (Higbie, 1935). The mass transfer coefficient is

$$k_{l,O_2} = \frac{2}{\sqrt{\pi}} \sqrt{\frac{D_{A,O_2}\, u_{rel}}{d}} \tag{53}$$

k_{l,O_2} is the mass transfer coefficient for oxygen, D_{A,O_2} the diffusion coefficient of oxygen in water, u_{rel} the magnitude of the relative velocity between the fluid and the bubble and d stands for the bubble diameter. As the bubbles are simulated using the Lagrangian method all these variables are directly available for every bubble. The oxygen saturation concentration in the liquid $c_{O_2,eq}$ is calculated using Henry’s law with the pressure sum p_{total} of headspace and hydrostatic pressure.

$$c_{O_2,eq} = c_w \frac{p_{total}\, x_{O_2}}{K_{H,O_2}^{px}} \tag{54}$$

where x_{O_2} is the mole fraction of oxygen in the gaseous phase, c_w is the concentration of water and K_{H,O_2}^{px} is the Henry coefficient for oxygen in water. We assume in the following simulations that the concentration of the oxygen in the bubble is constant and that the bubble diameter is not changing because of the oxygen dissolution. The total mass transfer $m_{O_2,trans}$ is calculated taking into account the bubble surface A_B, the time step length Δt, the molar mass M_{O_2}, and the current local oxygen saturation concentration and the local dissolved oxygen concentration in the liquid phase c_{O_2}.

$$m_{O_2,trans} = k_{l,O_2}\, A_B\, \Delta t\, (c_{O_2,eq} - c_{O_2}) M_{O_2} \tag{55}$$

After each time step the oxygen concentration in the gas bubbles and the respective bubble diameter is updated using the result of Equation (55). Note, that the model of Higbie (1935) applies Fick's second law of diffusion, while Equation (55) applies Fick's first law. However, this simplification is made only for processing to the next time step, and after that the gas phase state is updated. This is, the present approach represents a discretization of the instationary mass transfer process. Moreover, a simple biological model was implemented. The microorganisms are assumed to be homogeneously distributed in the reactor, converting the constantly fed substrate (e. g. a carbohydrate) and the dissolved oxygen into carbon dioxide. A simplified biological model is applied for illustrating the abilities of the code and which assumes metabolic activity of the microorganisms when both the substrate and the dissolved oxygen concentration are above a predefined threshold. The excreted carbon dioxide is then taken up by the passing air bubbles. To model the uptake of carbon dioxide again Higbie‘s penetration theory (Higbie, 1935) is used.

$$k_{l,CO_2} = \frac{2}{\sqrt{\pi}} \sqrt{\frac{D_{A,CO_2}\, u_{rel}}{d}} \tag{56}$$

k_{l,CO_2} denotes the mass transfer coefficient for carbon dioxide, D_{A,CO_2} the diffusion coefficient of carbon dioxide. The initial concentration of carbon dioxide in the gas bubbles is assumed to be zero, which results in an initial carbon dioxide saturation concentration in the fluid of $c_{CO_2,eq}$ of zero.

The resulting equation for the mass transfer of carbon dioxide $m_{CO_2,trans}$ is:

$$m_{CO_2,trans} = k_{l,CO_2}\, A_B\, \Delta t\, (-c_{CO_2}) M_{CO_2}. \quad (57)$$

M_{CO_2} is the molar mass of carbon dioxide. As described above for oxygen, the carbon dioxide concentration in the gas phase as well as the diameter of the gas bubbles are updated using the result of Equation (57) after each time step. Accordingly, the saturation concentration of carbon dioxide is updated.

4.3 Results and discussion

4.3.1 Performance and validation

To determine if the newly developed algorithm is fast and efficient enough, it was compared to a common particle-based solver. Since the latter guarantees non-negativity of the species concentrations and the absence of mass loss, it is a valid way to assess the new solver. For the 150 l reactor, one million particles were used to describe the distribution of the species inflow. 725 MB were required by the transport solver, and the calculation wall time was 61.6 % longer than that of the two-phase-flow simulation.

The new transport solver had a constant number of nodes of 1.08 million and the memory requirement of 161 MB throughout the entire simulation. Since the simulation was executed on two GPUs, the calculation of the species transport was partly absorbed by the waiting time required by the two-phase-flow algorithm. Hence, the simulation wall time was only 0.57 % longer. When the simulation was performed on one GPU, the wall time increased by 12.7 %. These data show the advantages of the new solver in terms of memory requirements and calculation efficiency.

The new species transport solver was tested via the Gaussian hill transport, which shows the numerical diffusion by plotting the maximum value of the hill over the transported distance.

The initial distribution of species was set to the Gaussian hill with a standard distribution of 2.84 times the node distance. The Gaussian hill's peak value was transported correctly, regardless of the setting of the macroscopic fluid velocities. However, the shape of the Gaussian hill broadened during the transport due to numerical diffusion. The decay of the maximum value of the Gaussian hill was 53 % after 10 cell widths, which is superior to the Upwind algorithm (21 % of the maximum value after 10 cell widths) but inferior to the QUICKEST algorithm of Vested et al. 1992 that preserved 76 % of the maximum value of the Gaussian hill.

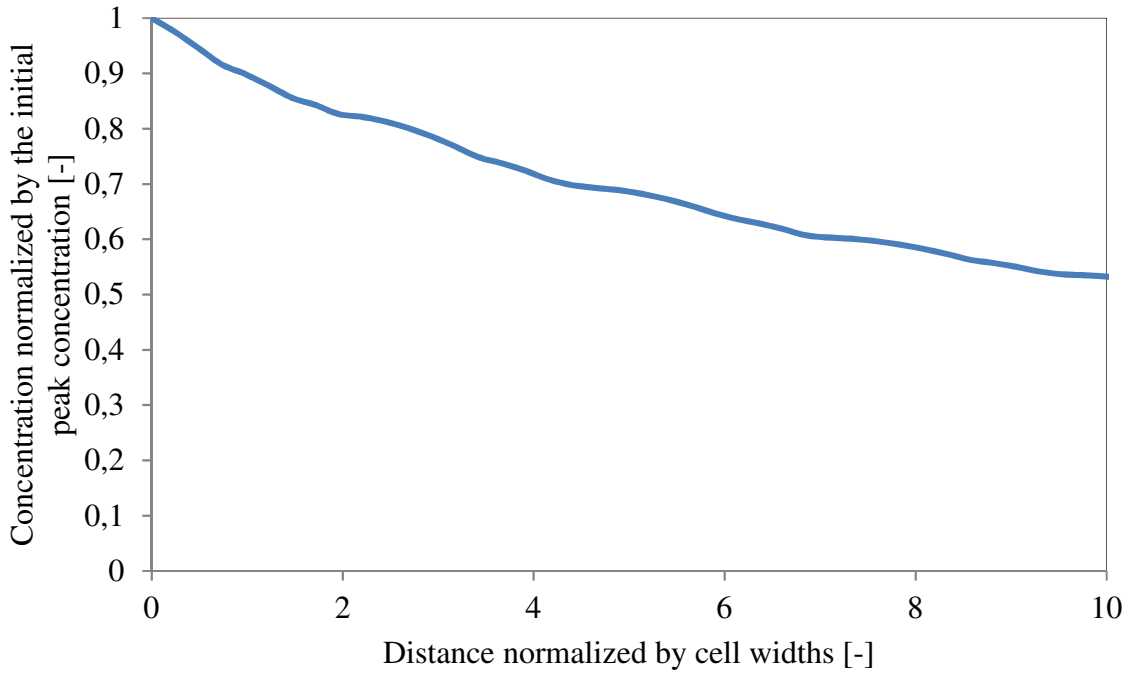

Figure 21: Diagram of peak value of the Gaussian hill transport

Reducing this numerical diffusion is subject to further research.

4.3.2 Gassed stirred tank bioreactor

The next step was to assess the new species transport solver in the full reactor setup with static and moving boundaries. During the test, no mass loss and no negative concentration values were observed. Figure 22 shows the distribution of the substrate injected at the injection site at a height of 0.9 m, a radial distance to the impeller axis of 0.15 m and an angle of 45° to the baffle.

The Figure shows that the substrate is first drawn axially from the top into the top Rushton turbine and then, distributed radially into the quadrant between the baffles where the outlet is located.

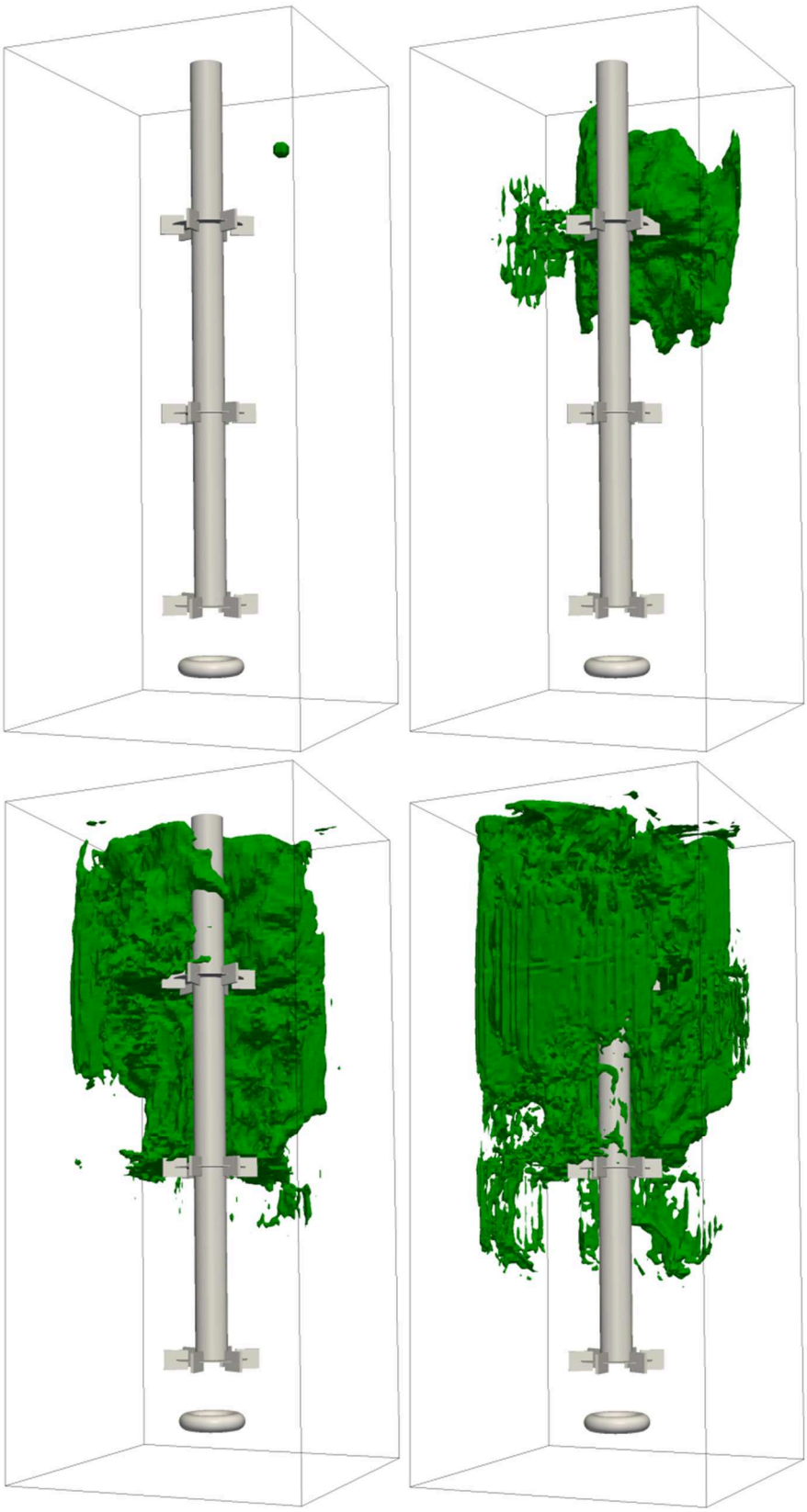

Figure 22: *Substrate distribution in the reactor, iso-surfaces for 0.247 g/m³ substrate per cell, after 0.04 s, 2.64 s, 4.16 s and 6.28 s.*

Interestingly, the substrate did not fill the entire horizontal section (i.e. all four quadrants) of the top stirrer compartment before being transported to the stirrer below. The bottom left part of Figure 22 indicates that the compartment between the baffles below the inlet was filled with the substance before the top stirrer's compartment was completely filled in the radial direction. The bottom right Figure shows a typical distribution pattern for multiple Rushton turbines. A portion of the lower vortex of the upper impeller is pulled downwards along the axis to the impeller below. At the lower impeller, the substrate is distributed via its upper and lower vortexes. This observation leads to the conclusion that a radial distribution of the injection sites (e.g. one inlet in every quadrant) can be advantageous for the speed of the substance distribution. Alternatively, the injection might be located next to the stirrer blades to provide a rapid radial distribution.

4.3.3 Single and split feed comparison

To compare single- and split-feeding strategies of the substrate, three simulations were conducted based on a standard 150l reactor. Inlet A is located above the top (third) stirrer in an area where only slow dispersion can be expected (case shown above). Inlet B is located at a height of 0.4 m, at a radial distance to the stirrer axis of 0.15 m and at a 45° angle to the same baffle as Inlet A. For each of these inlets, one simulation with a discharge rate of 0.02 kg/h was performed. Additionally, a simulation was conducted in which both inlets distributed half of the previous discharge rate (0.01 kg/h). To compare the feeding effectiveness, cells containing more than 0.5 mg/m³ of substrate were counted every 0.2 s (Figure 23).

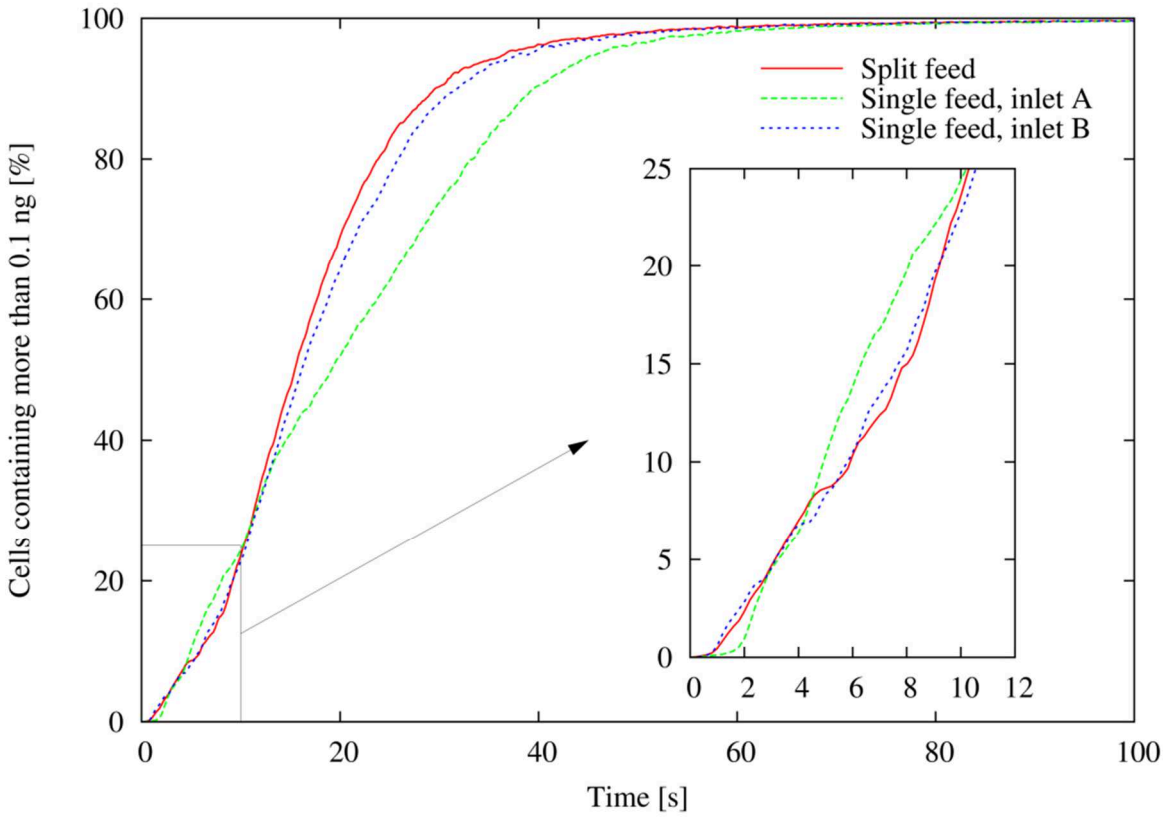

Figure 23: Comparison between split feed and single feed at two injection sites

As expected, mixing at Inlet A is slower than that that based on Inlet B, except for seconds 4-10 shown in the inset diagram. The reason is a better dispersion at inlet B, which leads to more cells containing the substance but fewer cells with the substance mass above the threshold. With 80% of all cells reaching the threshold, feeding via Inlet A takes 39.8% longer than split feeding and 29.0% longer than feeding via Inlet B.

Interestingly, split feeding offers only a small benefit (8.4%) in terms of reaching the 80% mixing time (80% Split: 23.85 s, inlet A: 25.85 s, B: 33.35 s), indicating that the placement of the inlets is more important than single or split feeding.

4.3.4 Oxygen distribution

Figure 24 shows the iso-surfaces of 5 mg/m³ of dissolved oxygen at 8 s, 12 s, 20 s and 28 s. The simulation is started with a water-filled reactor. It is assumed that no dissolved oxygen is present at the start of the simulation. The air flow through the ring sparger is started with the start of the simulation.

An unexpected phenomenon was observed in all four cases. Oxygen concentration was much higher in the downward facing vortices of the Rushton turbines, which can be explained by the dependence of k_l on the relative velocity between gas and liquid phase. The value of k_l directly depends on the value of u_{rel}. In the upward vortices, this value is low since the bubbles follow the macroscopic fluid velocity. However, in the downward vortex, the velocity of the bubbles caused by the drag is reduced by the buoyancy force, which points in the upward vertical direction. This causes a higher relative velocity u_{rel} and a higher value of k_l, which leads to a higher mass transfer rate of oxygen from the bubble to the fluid. Moreover, the concentration of dissolved oxygen in the vortex in the middle stirrer is higher than in the vortex at the bottom Rushton turbine. This is due to the bubble diameter that is indirectly related to the value of k_l and the value of bubble surface. Hence, the mass transferred increases significantly after the bubbles are broken up by the impeller.

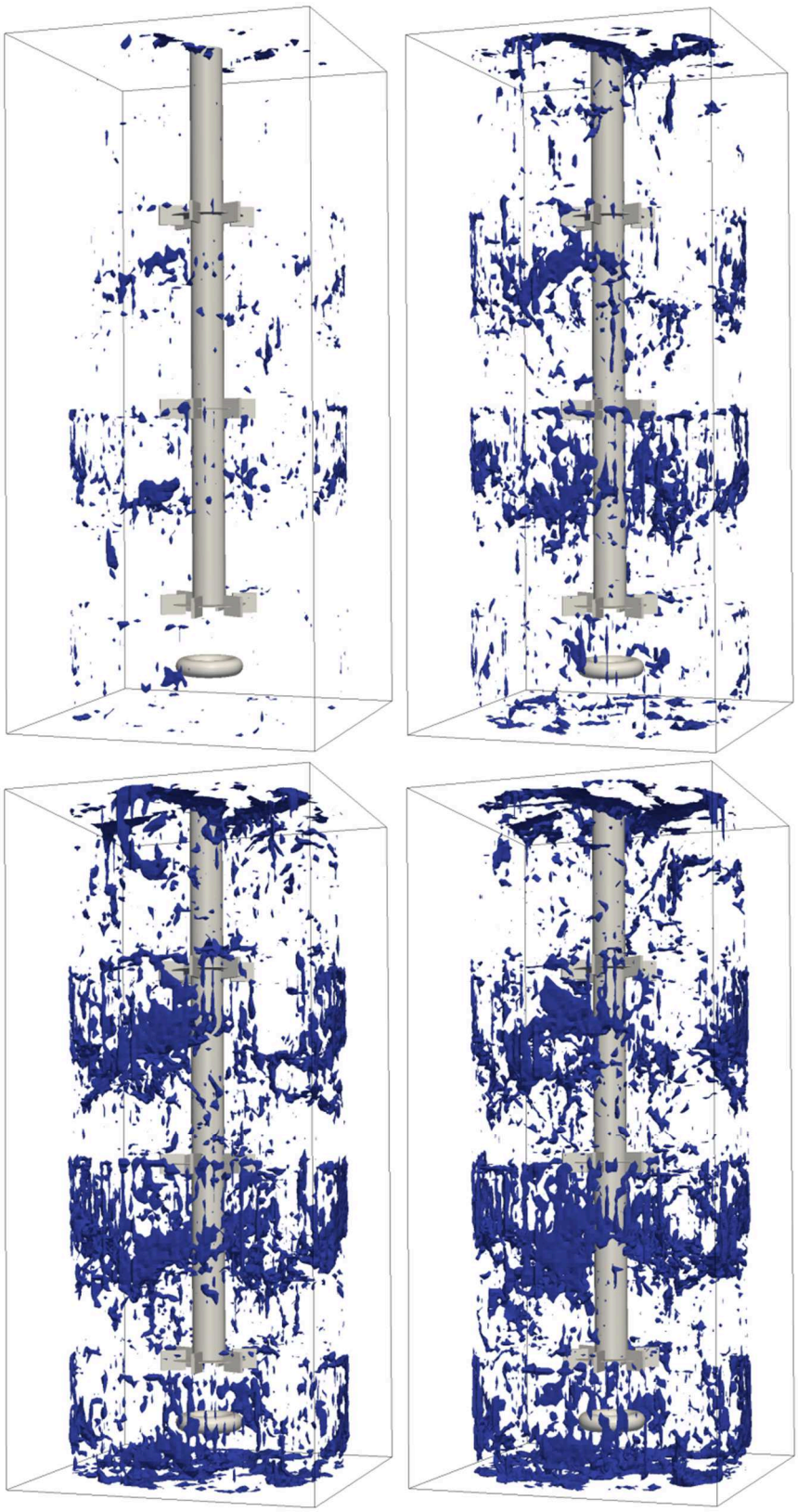

Figure 24 Oxygen distribution in reactor, iso surface for 5 mg/m³ of oxygen per cell, after 8 s, 12 s, 20 s and 28 s.

4.3.5 Biological model

The local concentration fields of substrate and dissolved oxygen can be used as the basis for implementing a biological model. We chose a simple model to demonstrate that all the necessary input and output fields are available and functioning. Implementing more sophisticated reaction models is straightforward and typically involves incorporating algebraic equations, lookup tables or empirical coefficients to simulate the behavior of microorganisms.

The present simple model compares the concentrations of dissolved oxygen and substrate with a threshold value. If the concentrations of both species are above the threshold, both concentrations are lowered, and the corresponding amount of dissolved carbon dioxide is produced. Since Inlet A at the top is used for feeding the substrate (Figure 25), the first active sites of the biological model are located at the top of the reactor, where the dissolved oxygen accumulates at the downward facing vortex of the top stirrer due to a higher mass transfer of oxygen, as described in Section 4.3.4.

The model shows the intersections between the iso surfaces of the dissolved oxygen and the substrate, which can be used for a first assessment of inlet positions or stirrer types and –heights in the reactor design process. For further insight on the production rate and the efficiency of different bioreactors configurations or operating parameter can be gained with the use of advanced models for the microbial metabolism like the Michaelis–Menten type of kinetics. The implementation is straightforward as all necessary input parameters required for the species transport solver and the biological model are available.

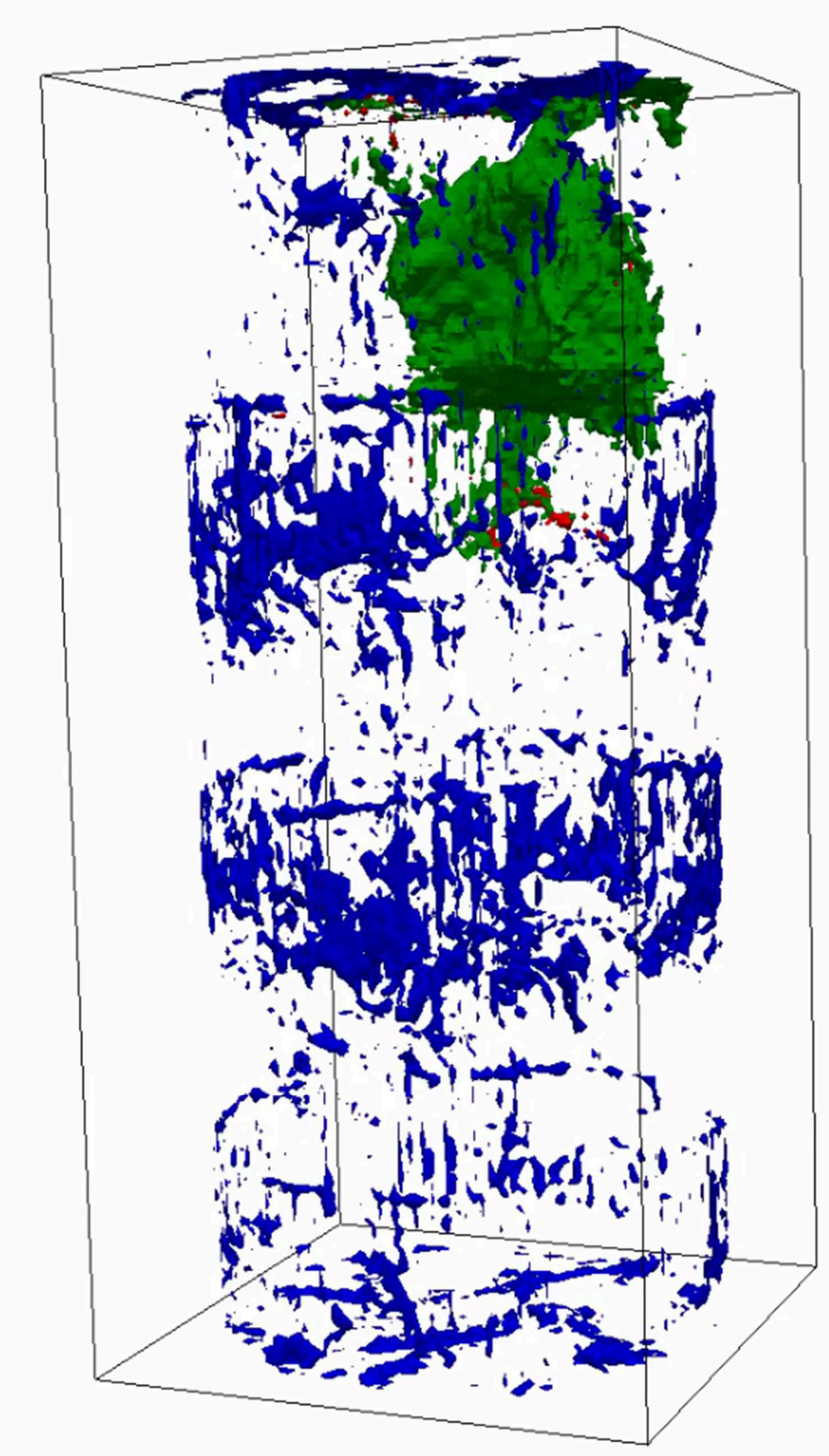

Figure 25: O_2 (blue), substrate (green) and CO_2 (red) distribution with the simple biological mode

4.4 Conclusions

A new method of simulating the advective species transport based on the lattice Boltzmann model was proposed. By using the artificial case of a resting fluid where a certain amount of species is added to one cell via the source term, the fast diffusion-like distribution of parts of the species mass became obvious. When a macroscopic fluid velocity is present, the dot product in the equilibrium distribution functions leads to unphysical transport of species mass in the opposite direction of the fluid flow. Additionally, the collision step preserves the direction of the species' origin, which is unphysical since the dissolved species is assumed to have no inertia and its advection can only depend on the direction of the macroscopic fluid flow.

The new method proposed in the present thesis sums up all of the species distribution vectors and the source term, removing all information of the species origin. Fractions of the species sum in the cell are assigned to the distribution vectors based on the macroscopic fluid velocities. The algorithm is restarted after the streaming step.

The simulation speed of this species transport solver is very high and negligible when the Eulerian LBM fluid flow field and the Lagrangian bubble dynamics are applied for the calculation in a dual GPU setup. The boundary conditions of the fluid flow field can be reused for the transport solver. To model the dissolution of oxygen from the gas bubbles and their uptake of carbon dioxide, Higbie's penetration theory was used.

The new solver can serve as a basis to include routines that limit numerical diffusion and to implement models for the diffusion of transported substance. The simple biological model can be enhanced by adding advanced algorithms that account for the microbial response to varying concentrations. The bioreactor design process can be aided by virtual experiments, including the biological activity.

4.5 Nomenclature

A_B – bubble surface, m²

C – stirrer height from bottom, m

c_{O_2} – oxygen concentration at the bubble location, mol/m³

$c_{O_2,bubble}$ – oxygen concentration in the bubble, mol/m³

D – stirrer diameter, m

d – bubble diameter, m

D_A – diffusion coefficient, m²/s

$\vec{e}_a$ – spatial direction

K_H – Henry coefficient, mol/(m³ Pa)

H – reactor height, m

k_l – mass transfer coefficient, m/s

μ – viscosity, Pa·s

m – mass species in a cell, kg

M_{O_2} – molar mass of oxygen, kg/mol

$m_{O_2,trans}$ – total mass transfer of oxygen, kg

m_S – species mass source, kg

p_{total} – sum of headspace and hydrostatic pressure, Pa

ρ – fluid density, kg/m³

s – gas sparger height, m

T_a – statistical distribution function

T – reactor diameter, m

Δt – time step length, s

u_{rel} – magnitude of the relative velocity between fluid and bubble, m/s

$\vec{u}_f$ – macroscopic fluid velocity, m/s

x_{O_2} – mole fraction of oxygen in the gaseous phase

4.6 References

Bailey P, Myre J, Walsh SDC, Lilja DJ, Saar MO. 2009. Accelerating Lattice Boltzmann Fluid Flow Simulations Using Graphics Processors. *2009 Int. Conf. Parallel Process.*:550–557.

Bhatnagar PL, Gross EP, Krook M. 1954. A model for collision process in gases. I. Small amplitude process in charged and neutral one-component system. *Phys. Rev.* **94**:511.

Bin D, Bao-Chang S, Guang-Chao W. 2005. A New Lattice Bhatnagar–Gross–Krook Model for the Convection–Diffusion Equation with a Source Term. *Chinese Phys. Lett.* **22**:267–270.

Camas BS. 2008. Lattice Boltzmann Modeling for Mass Transport Equations in Porous Media. **2002**.

Chai Z, Zhao TS. 2013. Lattice Boltzmann model for the convection-diffusion equation. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **87**:1–15.

Chai Z, Zhao TS. 2014. Nonequilibrium scheme for computing the flux of the convection-diffusion equation in the framework of the lattice Boltzmann method. *Phys. Rev. E* **90**:13305.

Chen S, Doolen GD. 1998. Lattice Boltzmann Method for Fluid Flows. *Annu. Rev. Fluid Mech.* **30**:329–364.

Chopard B, Falcone JL, Latt J. 2009. The lattice Boltzmann advection-diffusion model revisited. *Eur. Phys. J. Spec. Top.* **171**:245–249.

Devkota BH, Imberger J. 2009. Lagrangian modeling of advection-diffusion transport in open channel flow. *Water Resour. Res.* **45**:1–14.

Eggels JGM, Somers J a. 1995. Numerical simulation of free convective flow using the lattice-Boltzmann scheme. *Int. J. Heat Fluid Flow* **16**:357–364.

Gebäck T, Heintz A. 2013. A Lattice Boltzmann Method for the Advection-Diffusion Equation with Neumann Boundary Conditions. *Commun. Comput. Phys.* **15**:487–505.

Ginzburg I. 2005a. Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation. *Adv. Water Resour.* **28**:1171–1195.

Ginzburg I. 2005b. Generic boundary conditions for lattice Boltzmann models and their application to advection and anisotropic dispersion equations. *Adv. Water Resour.* **28**:1196–1216.

Guo Z, Shi B, Zheng C. 2002a. A coupled lattice BGK model for the Boussinesq equations. *Int. J. Numer. Methods Fluids* **39**:325–342.

Guo Z, Zheng C, Shi B. 2002b. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **65**:1–6.

Harten A. 1997. High Resolution Schemes for Hyperbolic Conservation Laws. *J. Comput. Phys.* **135**:260–278.

Hartmann H, Derksen JJ, van den Akker HEA. 2006. Mixing times in a turbulent stirred tank by means of LES. *AIChE J.* **52**:3696–3706.

Higbie R. 1935. The role of a pure gas into a still liquid during short priod of exposure. . Book *Transm.Am.Inst.Chem.Engr.* **35**:365–373.

Hu G, Celik I. 2008. Eulerian–Lagrangian based large-eddy simulation of a partially aerated flat bubble column. *Chem. Eng. Sci.* **63**:253–271.

Karimi S, Nakshatrala KB. 2016. Do Current Lattice Boltzmann Methods for Diffusion and Advection-Diffusion Equations Respect Maximum Principle and the Non-Negative Constraint? *Commun. Comput. Phys.* **20**:374–404.

Ladd A. 1994. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *J. Fluid Mech.* **211**:285–309.

Luo H, Svendsen HF. 1996. Theoretical Model for Drop and Bubble Breakup in Turbulent Dispersions. *AIChE J.* **42**:1225–1233.

Obrecht C, Kuznik F, Tourancheau B, Roux JJ. 2013a. Scalable lattice Boltzmann solvers for CUDA GPU clusters. *Parallel Comput.* **39**:259–270.

Obrecht C, Kuznik F, Tourancheau B, Roux J-J. 2011. A new approach to the lattice Boltzmann method for graphics processing units. *Comput. Math. with Appl.* **61**:3628–3638.

Obrecht C, Kuznik F, Tourancheau B, Roux J-J. 2013b. Multi-GPU implementation of the lattice Boltzmann method. *Comput. Math. with Appl.* **65**:252–261.

Perko J, Patel R a. 2014. Single-relaxation-time lattice Boltzmann scheme for advection-diffusion problems with large diffusion-coefficient heterogeneities and high-advection transport. *Phys. Rev. E* **89**:53309.

Ponce Dawson S, Chen S, Doolen GD. 1993. Lattice Boltzmann computations for

reaction-diffusion equations. *J. Chem. Phys.* **98**:1514.

Ponoth SS, Mclaughlin JB. 2000. Numerical simulation of mass transfer for bubbles in water. **55**:1237–1255.

Qian Y, D'Humières D, Lallemand P. 1992. Lattice BGK models for Navier-Stokes equation. *EPL (Europhysics Lett.)* **17**:479–484.

Rasin I, Succi S, Miller W. 2005. A multi-relaxation lattice kinetic method for passive scalar diffusion. *J. Comput. Phys.* **206**:453–462.

Riaud A, Zhao S, Wang K, Cheng Y, Luo G. 2014. Lattice-Boltzmann method for the simulation of multiphase mass transfer and reaction of dilute species. *Phys. Rev. E* **89**:53308.

Shi B, Deng B, Du R, Chen X. 2008. A new scheme for source term in LBGK model for convection-diffusion equation. *Comput. Math. with Appl.* **55**:1568–1575.

Smagorinsky J. 1963. General circulation experiments with the primitive equations: I. The basic experiment. *Mon. Weather Rev.* **91**:99–164.

van der Sman RGM, Ernst MH. 2000. Convection-Diffusion Lattice Boltzmann Scheme for Irregular Lattices. *J. Comput. Phys.* **160**:766–782.

Sommerfeld M. 2003. Euler/Lagrange calculations of bubbly flows with consideration of bubble coalescence. *Can. J. Chem. Eng.* **81**:508–518.

Sungkorn R, Derksen JJ, Khinast JG. 2011. Modeling of turbulent gas–liquid bubbly flows using stochastic Lagrangian model and lattice-Boltzmann scheme. *Chem. Eng. Sci.* **66**:2745–2757.

Sungkorn R, Derksen JJ, Khinast JG. 2012a. Euler–Lagrange modeling of a gas–liquid

stirred reactor with consideration of bubble breakage and coalescence. *AIChE J.* **58**:1356–1370.

Sungkorn R, Derksen JJ, Khinast JG. 2012b. Modeling of aerated stirred tanks with shear-thinning power law liquids. *Int. J. Heat Fluid Flow* **36**:153–166.

Tölke J. 2008. Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA. *Comput. Vis. Sci.* **13**:29–39.

Vested HJ, Justesen P, Ekebjærg L. 1992. Advection-dispersion modelling in three dimensions. *Appl. Math. Model.* **16**:506–519.

Witz C, Khinast JG. 2015. Bioreactor simulation with CUDA. In: Schindler, F-P, Kraume, M, editors *Fortschritt-Berichte VDI* Reihe 3. Düsseldorf: VDI-Verlag, pp. 91–105.

Witz C, Treffer D, Hardiman T, Khinast J. 2016. Local gas holdup simulation and validation of industrial-scale aerated bioreactors. *Chem. Eng. Sci.* **152**:636–648.

Yoshida H, Nagaoka M. 2010. Multiple-relaxation-time lattice Boltzmann model for the convection and anisotropic diffusion equation. *J. Comput. Phys.* **229**:7774–7795.

Yu H, Girimaji SS, Luo L-S. 2005. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *J. Comput. Phys.* **209**:599–616.

Zhang R, Fan H, Chen H. 2011. A lattice Boltzmann approach for solving scalar transport equations. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **369**:2264–2273.

Zhang X, Crawford JW, Glyn Bengough A, Young IM. 2002. On boundary conditions in the lattice Boltzmann model for advection and anisotropic dispersion equation.

Adv. Water Resour. **25**:601–609.

.

5
Conclusions and Outlook

5.1 Summary

In this thesis a simulation code for industrial-size aerated and stirred bioreactors was developed. The calculation power of graphic cards was used to reduce the simulation times significantly. The combination of selected algorithms with graphical processing units has proven to be a viable way to render the usage of a first principle based simulation in the bioreactor design process possible. The three key elements of the code include the fluid flow field calculation, the simulation of the bubble movement and the modelling of the species transport:

- The D3Q19 lattice Boltzmann model was selected and implemented to model the fluid flow field. For the collision step the equilibrium distribution vectors are calculated using the Bhatnagar-Gross-Krook approximation. The highly turbulent flow inside the reactor is simulated with the large eddy turbulence model including the method of Smagorinsky (Smagorinsky, 1963) to handle the subgrid turbulence. To include the reactor geometry as well as the stirrer the modified bounce back boundary method of Ladd (Ladd, 1994) is used.
- The bubbles are organized in parcels which renders the use of stochastic collision, breakup and coalescence models possible. To calculate the fluid forces on the bubble the fluid flow field variables are interpolated from the fluid nodes to the bubble position by a fourth-order mapping function (Deen et al., 2002). The bubble velocity change within one time step is calculated by solving the reduced equations of motion (Hu and Celik, 2008).

The local statistics around a bubble parcel determines the collision, breakup and coalescence probability (Luo and Svendsen, 1996; Sommerfeld, 2003).

- In chapter four a species transport algorithm was developed on basis of the lattice Boltzmann method. The solver avoids the problems that are commonly faced when using the lattice Boltzmann model for species transport: negative concentration values and artificial inertia of the dissolved species. The resulting species transport algorithm integrates perfectly in the multiphase simulation code and barely reduces the speed of the code execution.

5.2 Achievements and Conclusions

The main goals that were achieved are:

- The large scale Euler-Lagrangian aerated and stirred reactor simulation code was created using GPU-optimized algorithms. The algorithms were validated using literature data (chapter two) and data from experiments with a conductivity sensor (chapter three).
- The carefully chosen algorithms executed on the graphic cards reduced the simulation time for pilot size multiphase reactors from months to weeks and enabled large scale reactor simulations up to 250 m³.
- The newly developed species transport solver has proven to be highly efficient and well adapted to GPUs (chapter four). The solver was already applied for simulating the transient distribution of a substance feed in the reactor, for comparing different feeding points and for exhibiting the differences of single and split feeding of the substances without increasing the simulation time significantly.

5.3 Future Work

The simulation code can serve as a basis to include modules and algorithms that extend the abilities of the code:

- The volume averaged Navier Stokes equations can be derived for three dimensions for the application in the lattice Boltzmann method (two dimensions in (Blais et al., 2015)) to be able to simulate regions with high disperse phase fractions more accurately.
- Bioreactors have many different vessel shapes and installations. Only some of them are currently included in the code. Klöpper or hemispherical heads could be implemented as well as large scale gas spargers and heat exchangers.
- The simple biological model described in chapter four can be replaced with more advanced models which take the dynamics of the response of microorganisms to starving and saturation into account. Additionally, the particle strain of microorganisms due to excessive shear rate could be modeled.
- The temperature profile inside the reactor is important since the metabolic processes are exothermal. A module which includes the heat generation of the microorganisms and the cooling effect of the immersed heat exchangers could generate insight in the temperature distribution inside the reactor.

5.4 References

Blais B, Tucny J-M, Vidal D, Bertrand F. 2015. A conservative lattice Boltzmann model for the volume-averaged Navier–Stokes equations based on a novel collision operator. *J. Comput. Phys.* **294**:258–273.

Deen NG, Solberg T, Hjertager BH. 2002. Flow generated by an aerated Rushton impeller: two-phase PIV experiments and numerical simulations. *Can. J. Chem. Eng.* **80**:1–15.

Hu G, Celik I. 2008. Eulerian–Lagrangian based large-eddy simulation of a partially aerated flat bubble column. *Chem. Eng. Sci.* **63**:253–271.

Ladd A. 1994. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *J. Fluid Mech.* **211**:285–309.

Luo H, Svendsen HF. 1996. Theoretical Model for Drop and Bubble Breakup in Turbulent Dispersions. *AIChE J.* **42**:1225–1233.

Smagorinsky J. 1963. General circulation experiments with the primitive equations: I. The basic experiment. *Mon. Weather Rev.* **91**:99–164.

Sommerfeld M. 2003. Euler/Lagrange calculations of bubbly flows with consideration of bubble coalescence. *Can. J. Chem. Eng.* **81**:508–518.