

# **A Consistent Large Eddy Approach for Lattice Boltzmann Methods and its Application to Complex Flows**

Vom Fachbereich Maschinenbau und Verfahrenstechnik  
der Technischen Universität Kaiserslautern  
zur Verleihung des akademischen Grades

**Doktor-Ingenieur (Dr.-Ing.)**  
genehmigte  
**Dissertation**

von

Herrn

Dipl.-Ing. Andreas Schneider  
aus Landstuhl

Tag der mündlichen Prüfung: 23. März 2015  
Dekan: Prof. Dr.-Ing. Christian Schindler  
Vorsitzender: Prof. Dr.-Ing. Sergiy Antonyuk  
Berichterstatter: Prof. Dr.-Ing. Martin Böhle  
Prof. Dr.-Ing. habil. Uwe Janoske

D 386



---

## Vorwort

---

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Strömungsmechanik und Strömungsmaschinen der Technischen Universität Kaiserslautern.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Martin Böhle für die steten Anregungen, Diskussionen und Ratschläge, die wesentlich zum Gelingen dieser Arbeit beigetragen haben.

Ebenfalls bedanke ich mich bei Herrn Prof. Dr.-Ing. habil. Uwe Janoske für die Übernahme des Koreferats und bei Herrn Prof. Dr.-Ing. Sergiy Antonyuk für die Übernahme des Vorsitzes der Prüfungskommission.

Den Kolleginnen und Kollegen am Lehrstuhl für Strömungsmechanik und Strömungsmaschinen danke ich für das angenehme, sehr kollegiale Arbeitsklima und für viele freundschaftliche Gespräche. Meinen studentischen Hilfskräften und Studien- und Diplomarbeitern danke ich für ihre tatkräftige Mithilfe.

Größten Dank spreche ich meinem Kollegen und Freund Daniel Conrad für die tolle Zusammenarbeit und die Unterstützung in allen Lebenslagen aus.

Ein großer Dank gilt auch meiner Familie und meinen Freunden. Meinen Eltern Werner und Sonja dafür, dass sie mir eine hervorragende Ausbildung ermöglicht haben und mich bei meinen Entscheidungen und meinem Handeln immer unterstützt haben. Meinen Brüdern Mathias und Marcel für ihr Vertrauen in mich und für viele Ratschläge. Allen meinen Freunden danke ich für ihren Beistand und ihre Unterstützung auch in schwierigen Zeiten.

Kaiserslautern, im April 2015

Andreas Schneider





---

# Contents

---

<b>Vorwort</b>	<b>iii</b>
<b>Abstract</b>	<b>ix</b>
<b>Kurzfassung auf Deutsch</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Scope of this Thesis . . . . .	2
1.3 Physical Description of Fluid Flows . . . . .	3
1.4 Mathematical Notation . . . . .	4
<b>2 Kinetic Theory</b>	<b>7</b>
2.1 The Distribution Function . . . . .	7
2.2 The Boltzmann Equation . . . . .	9
2.3 The Equilibrium Distribution Function . . . . .	10
2.4 BGK Approximation . . . . .	11
2.5 The Chapman-Enskog Expansion . . . . .	12
<b>3 The Lattice Boltzmann Method</b>	<b>15</b>
3.1 The Lattice Boltzmann Equation . . . . .	15
3.1.1 Discretization of Velocity Space . . . . .	16
3.1.2 Discretization in Space and Time . . . . .	18
3.2 Discrete Phase Space Model . . . . .	21
3.3 Collision Operators . . . . .	24
3.3.1 Single Relaxation Time Model . . . . .	24

3.3.2	Multiple Relaxation Time Model . . . . .	25
3.4	Treatment of External Forces . . . . .	27
3.4.1	Single Relaxation Time Model . . . . .	27
3.4.2	Multiple Relaxation Time Model . . . . .	28
3.5	Boundary Conditions . . . . .	29
3.5.1	Equilibrium Schemes . . . . .	29
3.5.2	Periodic Boundary Conditions . . . . .	29
3.5.3	Bounce Back Schemes . . . . .	30
3.5.4	Bouzidi Scheme . . . . .	32
3.5.5	Yu Scheme . . . . .	33
3.5.6	Mei Scheme . . . . .	35
3.5.7	Shear Boundaries . . . . .	35
3.6	Initial Conditions . . . . .	41
3.6.1	Analytical Schemes . . . . .	41
3.6.2	Iterative Schemes . . . . .	41
3.7	Local Grid Refinement . . . . .	41
3.7.1	General Concept . . . . .	42
3.7.2	SRT Scaling . . . . .	46
3.7.3	MRT Scaling . . . . .	47
3.7.4	Nested Time Stepping . . . . .	48
3.8	Accuracy and Stability . . . . .	48
3.8.1	Accuracy Issues . . . . .	48
3.8.2	Stability Issues . . . . .	50
<b>4</b>	<b>Large Eddy Simulation for Lattice Boltzmann Methods</b>	<b>53</b>
4.1	Principles of Large Eddy Simulation . . . . .	53
4.2	Application to the Lattice Boltzmann Method . . . . .	56
4.2.1	Filtered Lattice Boltzmann Equation . . . . .	57
4.2.2	Subgrid Scale Models . . . . .	59
4.3	Boundary Conditions for Large Eddy Simulation . . . . .	61
4.3.1	Wall Boundary Conditions . . . . .	61
4.3.2	Inlet Boundary Conditions . . . . .	69
4.4	Grid Refinement for Large Eddy Simulation . . . . .	74
4.5	Practical Aspects for Large Eddy Simulation . . . . .	75
<b>5</b>	<b>Implementation Aspects of the CFD Package SAM-Lattice</b>	<b>77</b>
5.1	Conceptual Aspects . . . . .	77
5.2	Preprocessor - SamGenerator . . . . .	78
5.2.1	Input Data . . . . .	80
5.2.2	Discretization of Grid Level 0 . . . . .	80
5.2.3	Local Grid Refinement . . . . .	81

5.2.4	Lattice Definition . . . . .	82
5.2.5	Output Data . . . . .	83
5.3	Solver - SamSolver . . . . .	83
5.3.1	Input Data . . . . .	84
5.3.2	Preprocessing . . . . .	86
5.3.3	Solving . . . . .	86
5.3.4	Output Data . . . . .	87
<b>6</b>	<b>Verification Cases</b>	<b>89</b>
6.1	Core Components . . . . .	90
6.1.1	Beltrami Flow . . . . .	90
6.1.2	Taylor-Couette Flow . . . . .	95
6.1.3	Couette Flow . . . . .	99
6.1.4	Jeffery-Hamel Flow . . . . .	104
6.2	LES Components . . . . .	108
6.2.1	Wall Resolved Approach . . . . .	108
6.2.2	Wall Modeled Approach . . . . .	113
6.2.3	Synthetic Eddy Method . . . . .	119
<b>7</b>	<b>Pump Intake Flows - Complex Validation Cases</b>	<b>123</b>
7.1	Pump Intakes . . . . .	123
7.2	Free Surface Intake . . . . .	128
7.3	Pressurized Intake . . . . .	137
<b>8</b>	<b>Conclusions and Future Work</b>	<b>153</b>
<b>A</b>	<b>D3Q19 MRT Model</b>	<b>155</b>
	<b>Nomenclature</b>	<b>159</b>
	<b>List of Figures</b>	<b>163</b>
	<b>List of Tables</b>	<b>167</b>
	<b>Bibliography</b>	<b>169</b>
	<b>Curriculum Vitae</b>	<b>179</b>



---

## Abstract

---

Lattice Boltzmann Methods have shown to be promising tools for solving fluid flow problems. This is related to the advantages of these methods, which are among others, the simplicity in handling complex geometries and the high efficiency in calculating transient flows. Lattice Boltzmann Methods are mesoscopic methods, based on discrete particle dynamics. This is in contrast to conventional Computational Fluid Dynamics methods, which are based on the solution of the continuum equations. Calculations of turbulent flows in engineering depend in general on modeling, since resolving of all turbulent scales is and will be in near future far beyond the computational possibilities. One of the most auspicious modeling approaches is the large eddy simulation, in which the large, inhomogeneous turbulence structures are directly computed and the smaller, more homogeneous structures are modeled.

In this thesis, a consistent large eddy approach for the Lattice Boltzmann Method is introduced. This large eddy model includes, besides a subgrid scale model, appropriate boundary conditions for wall resolved and wall modeled calculations. It also provides conditions for turbulent domain inlets. For the case of wall modeled simulations, a two layer wall model is derived in the Lattice Boltzmann context. Turbulent inlet conditions are achieved by means of a synthetic turbulence technique within the Lattice Boltzmann Method.

The proposed approach is implemented in the Lattice Boltzmann based CFD package SAM-Lattice, which has been created in the course of this work. SAM-Lattice is feasible of the calculation of incompressible or weakly compressible, isothermal flows of engineering interest in complex three dimensional domains. Special design targets of SAM-Lattice are high automatization and high performance.

Validation of the suggested large eddy Lattice Boltzmann scheme is performed for pump intake flows, which have not yet been treated by LBM. Even though, this numerical method is very suitable for this kind of vortical flows in complicated domains. In general, applications of LBM to hydrodynamic engineering problems are rare. The results of the pump intake validation cases reveal that the proposed numerical approach is able to represent qualitatively and quantitatively the very complex flows in the intakes. The findings provided in this thesis can serve as the basis for a broader application of LBM in hydrodynamic engineering problems.



---

## Kurzfassung auf Deutsch

---

Seit einiger Zeit zeigen die Tendenzen im Ingenieurwesen, dass die Produktentwicklung und Herstellung zunehmend von rechnerbasierten Prozessen beeinflusst bzw. dominiert werden. Entsprechende Programme werden unter dem Begriff Computer-Aided Technologies, abgekürzt CAx, zusammengefasst. Die Gründe für diese Tendenzen sind vielfältig, zu den treibenden Parametern sind natürlich die Reduktion von Entwicklungskosten und Zeiten zu zählen. Auf der Basis einer detaillierten und verlässlichen Produktauslegung, kann die Anzahl der notwendigen, teuren Prototypen und die damit verbunden experimentellen Untersuchungen signifikant reduziert werden. Der Einsatz von CAx Technologien ist jedoch direkt mit den verfügbaren Computerressourcen verknüpft, die in den letzten zwei Jahrzehnten unvorstellbar gewachsen sind. Natürlich erfordert der Einsatz von CAx, neben verfügbaren Rechnerressourcen, auch genaue und brauchbare numerische Verfahren für die entsprechenden Problemstellungen.

Eine Sparte von CAx ist die numerische Strömungsmechanik, gewöhnlich mit dem englischsprachigen Begriff Computational Fluid Dynamics (CFD) bezeichnet. Numerische Verfahren spielen in der Strömungsmechanik eine bedeutende Rolle, weil die beschreibenden Gleichungen für ziemlich alle Strömungsprobleme nicht analytisch lösbar sind. Ein tieferer Einblick bzw. tieferes Verständnis ist deshalb nur mit simulativen oder experimentellen Methoden möglich. Zurzeit finden vor allem finite Volumen Diskretisierungen der Navier-Stokes Gleichungen, welche die Strömung als Kontinuum beschreiben, Anwendung in der numerischen Strömungsmechanik. Diese Verfahren stellen hohe Anforderungen an die Qualität der verwendeten Gebietsdiskretisierung, was einen arbeitsintensiven Prozess, das sogenannte Preprocessing, mit sich bringt. Das Preprocessing erfordert hierbei viele manuelle Eingriffe des Ingenieurs und die Qualität der resultierenden Diskretisierung ist stark mit dessen Erfahrung verbunden. Die genannten Punkte erschweren die Standardisierung des Preprocessings und zeigen eine Notwendigkeit für Verfahren, die stark automatisierbar sind.

Neben dem oben genannten Kontinuumsansatz, können Strömungen auch durch eine mesoskopische Betrachtungsweise, mit Hilfe der Boltzmann Gleichung, beschrieben werden. In den letzten Jahren wurde ein numerisches Verfahren, die Lattice Boltzmann Methode, welches auf dem mesoskopischen Ansatz basiert, vorgestellt und vorangetrieben. Die aufkommende Be-

liebtheit dieser Methode im Ingenieurwesen ist mit ihren Vorteilen verbunden. Zu diesen zählen unter anderem die einfache Behandlung komplexer Geometrien, die hohe Effizienz des Verfahrens in der Berechnung zeitabhängiger Strömungen und die einfache Einbeziehung von verschiedenen physikalischen Modellen. Aufgrund des erstgenannten Punktes lässt sich ein stark automatisierter CFD Prozess realisieren, der keine manuelle Interaktion erfordert und somit die Anforderungen vieler Anwendungen erfüllt.

Die meisten Strömungen in Ingenieur Anwendungen sind turbulenter Art. Eine direkte Berechnung aller turbulenter Schwankungen ist im Allgemeinen nicht umsetzbar, aufgrund des erforderlichen Berechnungsaufwands. Deswegen werden verschiedene Modellierungsansätze verwendet, um eine Berechnung zu ermöglichen und den Einfluss der Turbulenz zu berücksichtigen. Ein solcher Ansatz ist die Grobstruktursimulation, gemeinhin als Large Eddy Simulation (LES) bekannt. In der Grobstruktursimulation werden die großen turbulenten Strukturen, die sehr inhomogen sind und stark von der jeweiligen Strömung abhängen, direkt berechnet und die kleinskaligen Strukturen modelliert. Die Modellierung der kleinskaligen Strukturen ist aus physikalischer Sicht einfacher, da diese homogener sind. Weitere Limitierungen sind bei hohen Reynoldszahlen, wie sie in realen Anwendungen üblich sind, durch die Auflösung der dünnen Wandgrenzschichten gegeben. In vielen Fällen ist es aus ingenieurtechnischer Sicht nicht tragbar und notwendig die Grenzschichten vollständig aufzulösen. Die Effekte nicht aufgelöster Grenzschichten müssen dann aber anhand eines sogenannten Wandmodells berücksichtigt werden. Grobstruktursimulation mit der Lattice Boltzmann Methode erfordert akkurate Modelle, die diese Anforderungen erfüllen.

In dieser Dissertation wird ein konsistentes Grobstrukturmodell für Lattice Boltzmann Methoden erarbeitet. Der Ansatz beinhaltet außer einem Feinstrukturmodell, das die kleinen Skalen abbildet, die entsprechenden Randbedingungen. Neben den klassischen Randbedingungen für wandaufgelöste Berechnungen wird ein zwei Schichtenwandmodell für nicht aufgelöste Grenzschichten im Lattice Boltzmann Kontext entwickelt. Die meisten Strömungen in den Ingenieur Anwendungen benötigen darüber hinaus turbulente Randbedingungen am Gebietseintritt. Zur Erzeugung der stochastischen, turbulenten Fluktuationen am Eintritt wird eine Methode aus dem Bereich der synthetischen Turbulenz, die Synthetic Eddy Methode, in das Lattice Boltzmann Verfahren eingeführt.

Für die Berechnung realer Strömungsprobleme, wird das vorgeschlagene Grobstrukturmodell in das CFD Paket SAM-Lattice integriert, welches im Rahmen dieser Arbeit entwickelt wurde. SAM-Lattice ermöglicht die Berechnung inkompressibler bzw. schwach kompressibler, isothermaler Strömungen in beliebigen, dreidimensionalen Geometrien. Der modulare Aufbau von SAM-Lattice, die Struktur und die Grundlagen der einzelnen Komponenten werden umfangreich erläutert.

Darüber hinaus werden ausführlich die theoretischen Grundlagen der Lattice Boltzmann Methode, wie sie in SAM-Lattice zur Anwendung kommt, erklärt. Diese Darstellung wird, neben Gründen der Vollständigkeit, aus einem didaktischen Grund vorgenommen. Dieser Teil der Arbeit soll im weiteren Studenten und Wissenschaftlern, die mit SAM-Lattice arbeiten, als



theoretisches Nachschlagewerk dienen. Die Lattice Boltzmann Methode wird hierzu, ausgehend von der kinetischen Theorie bzw. der Boltzmann Gleichung, abgeleitet. Die getroffenen Annahmen und Vereinfachungen während dieser Herleitung werden aufgezeigt und diskutiert, was die Möglichkeit eröffnet das Verfahren an diesen Stellen über die momentanen Grenzen hinaus zu erweitern.

Der Entwicklungsprozess jeglicher numerischer Methode erfordert Verifizierung und Validierung, was ebenfalls in dieser Arbeit durchgeführt wird. Verifizierung für das Software Paket und die verwendeten Modelle wird im laminaren Fall mittels analytischer Lösungen der Navier-Stokes Gleichungen vorgenommen. Mit Hilfe der Beltrami Strömung, der Taylor-Couette Strömung, der Couette Strömung im ebenen Kanal und im Kreisring, sowie der Jeffery-Hamel Strömung werden die akkurate Implementierung der Kernkomponenten, der verschiedenen Kollisionsoperatoren und der verschiedenen Randbedingungen nachgewiesen. Die Ergebnisse zeigen das theoretisch erwartete Verhalten: Das Verfahren produziert für transiente Strömungen zeitgenaue Lösungen und besitzt, bezogen auf die räumliche Diskretisierung, eine Genauigkeit zweiter Ordnung, solange die Randbedingungen ebenfalls diese Ordnung aufweisen. Die verwendete Gitterverfeinerungsstrategie hat für die Jeffery-Hamel Strömung gleichermaßen ihre Richtigkeit unter Beweis gestellt. Das vorgeschlagene LES basierte Turbulenzmodell und die darin verwendeten Komponenten bzw. Randbedingungen werden anhand der turbulenten Kanalströmung verifiziert, für die eine große und verlässliche Basis von direkten numerischen Simulationen zur Verfügung steht. Im Falle von wandaufgelösten Berechnungen wird eine exzellente Übereinstimmung zwischen den SAM-Lattice Ergebnissen und der Referenzlösung festgestellt. In den wandmodellierten Berechnungen kann ebenfalls eine gute Kongruenz zwischen dem Lattice Boltzmann Resultat und der Referenz gezeigt werden. Zusammen mit dem erfolgreichen Nachweis für das synthetische Turbulenzverfahren als Eintrittsrandbedingung, ist die Verifizierung des konsistenten Grobstrukturansatzes erbracht.

Um neue Anwendungsfelder für die Lattice-Boltzmann Methode und das vorgeschlagene Grobstrukturmodell im Bereich der Hydrodynamik zu erschließen und zu motivieren, wird das vorgestellte Programmpaket für die Strömungen in Einlaufkammern validiert. Die Strömungen in Einlaufkammern sind in der Regel sehr komplex, da sie stark wirbelbehaftet sind und darüber hinaus normalerweise in komplizierten Geometrien auftreten. Diese Art von Strömungen wurden bisher nicht mit der Lattice Boltzmann Methode behandelt, wenn auch die Methode sehr passende Eigenschaften hierfür aufweist. Einlaufkammern können prinzipiell in zwei Gattungen unterschieden werden: Kammern mit freien Oberflächen und geschlossene Kammern. Für jeden dieser Fälle wird zu Validierungszwecken ein repräsentativer Vertreter, der in der Literatur dokumentiert ist, berechnet. Im Falle der Kammer mit freier Oberfläche wird ein zusätzliches Modellierungskonzept verwendet und begründet, da SAM-Lattice momentan nur zur Berechnung einphasiger Strömungen eingesetzt werden kann. Trotz dieser weiteren Modellierung ist das entwickelte Verfahren in der Lage, die verschiedenen Wirbelstrukturen in der Einlaufkammer wiederzugeben und auch das zeitabhängige Wandern und Verschwinden bzw. Wiederauftreten der Wirbel zu reproduzieren. Quantitative Vergleiche der berechneten und gemessenen

Wirbelpositionen zeigen sehr gute Vereinbarkeit. Für die geschlossene Einlaufkammer stehen sehr detaillierte Messungen der Felder für Geschwindigkeit und turbulente kinetische Energie zur Verfügung, die eine umfassende qualitative und quantitative Validierung des entwickelten Verfahrens ermöglichen. Die Ergebnisse dieses zweiten Validierungsfalles zeigen erneut, dass die Grobstruktur-Lattice Boltzmann Methode im Stande ist, die sehr komplexe, stark wirbelbehaftete Strömung in der Einlaufkammer qualitativ und quantitativ darzustellen. Somit ist das entwickelte und vorgestellte Verfahren validiert und hat seine Anwendbarkeit auf komplexe Strömungen in komplizierten Geometrien unter Beweis gestellt.

Abschließend werden in dieser Arbeit Möglichkeiten und Vorschläge zur Weiterentwicklung der Lattice Boltzmann Methode bzw. des CFD Pakets SAM-Lattice gegeben.

# CHAPTER 1

---

## Introduction

---

*“The purpose of computing is insight, not numbers.”*

— Richard Hamming, 1962

### 1.1 Motivation

Since quite some time, trends in engineering show that product design and manufacturing are more and more influenced and defined by computational processes. The according tools are commonly combined under the term computer-aided technologies (CAx). Reasons for these trends are manifold. Of course the reduction of development costs and times in the design process belong to the main influencing parameters. By means of more detailed and more reliable preliminary product designs, the required number of cost and time intensive prototypes and the associated experimental testing is significantly reduced. The usage of CAx in engineering is enabled and related to the available computational resources, which increased tremendously in the last two decades. Naturally, a critical aspects of this evolution is the availability of accurate and feasible numerical schemes for the related problems.

A category of CAx are *Computational Fluid Dynamics* (CFD), which deal with the numerical calculation of fluid flows. Numerical techniques are of special importance in fluid dynamics, since the governing equations cannot be solved analytically for almost all problems of engineering interest. A deeper insight in the nature of flows can thus only be provided by simulation and experiment. The currently most widely employed CFD schemes are based on finite volume discretizations of the Navier-Stokes equations, which describe fluid flow at continuum level. These methods demand a high quality domain discretization, what causes a time intensive preprocessing with many manual interactions and is strongly relying on the experience of

the engineer. This complicates standardization of preprocessing and brings up the necessity for schemes, which allow highly automated processes.

Besides the continuum approach, fluid flows can be described on a mesoscopic level with the help of the Boltzmann equation. In recent years, a numerical technique based on this approach has been proposed and advanced: the Lattice Boltzmann Method (LBM). Meanwhile, the Lattice Boltzmann Method has gained popularity for solving fluid flow problems of engineering interest. This is promoted by the advantages of this scheme, which are among others the simplicity to handle complex geometries, the high efficiency in calculating transient flows and the natural inclusion of many physical models. By the first mentioned point a highly automated CFD process with nearly no manual interaction is enabled and meets the need of many applications.

Most flows in engineering applications are characterized by a turbulent nature. In general, a direct calculation of all turbulent structures is not feasible, due to the enormous computational demand. Thus, different modeling approaches are applied to account for turbulence. One such approach is the large eddy simulation, in which the large turbulent structures are directly computed and the smaller structures are modeled. A further limitation at high Reynolds numbers, which are common in engineering, is the point that it is in many cases not feasible to resolve the boundary layers down to the walls. Hence, the effects of the unresolved parts of the boundary layer must be considered by wall models in these instances. Large eddy simulation for LBM requires accurate models, which fulfill these demands. The efficiency of a numerical scheme gets a crucial factor in the light of the large eddy simulation. The more efficient a scheme, the more turbulent structures can be resolved with a given computational power. LBM, as a highly efficient numerical scheme, seems promising for this purpose.

From these physical and technical demands the motivation for this thesis is given and the scope and aims can be defined in the following.

## **1.2 Scope of this Thesis**

A novel, consistent large eddy approach for the Lattice Boltzmann Method is proposed in this thesis. Large eddy models have been proposed in the Lattice Boltzmann context, but a consistent approach, which includes besides the subgrid scale model accurate boundary conditions, is still missing. This is addressed by the proposed approach. It includes proper boundary treatment for wall resolved and wall modeled calculations and also provides conditions for turbulent domain inlets. In the case of wall modeled simulations, a two layer wall model for large eddy simulation with Lattice Boltzmann Methods is derived. An appropriate turbulent inlet condition is introduced to the Lattice Boltzmann Method by means of a synthetic turbulence technique. Thus, a complete and consistent large eddy model for the accurate calculation of technical flows in complex domains, based on the Lattice Boltzmann Method, is developed.

For application of the Lattice Boltzmann Method and the suggested large eddy approach to engineering flow problems, the CFD software package SAM-Lattice has been created in the

course of this work. SAM-Lattice aims at the calculation of incompressible or weakly compressible isothermal flows of engineering interest in arbitrary three dimensional domains. From an user point of view, primary requirements to the software are a highly automated preprocessing, with minimal manual interaction, and a fast solver, capable to calculate the aforementioned flows accurately. These demands are considered and realized in the development of SAM-Lattice. The CFD package is characterized by a modular structure and an object oriented programming concept in C++.

To expand the area of application of Lattice Boltzmann Methods, the proposed large eddy scheme is applied to and validated for pump intake flows, a special field of hydraulic engineering with many links to turbo machinery. Pump intake flows have not yet been treated by LBM, although LBM is very suitable for this kind of vortical flows in complex domains. In general, applications of LBM to hydrodynamic engineering problems are rare. So, the research provided here should serve as the basis for a broader application of LBM in hydrodynamic engineering problems.

Besides these ambitions, this thesis also addresses educational purposes. It should serve as a theoretical explanation and documentation of the Lattice Boltzmann Method and its utilization in SAM-Lattice for students and researchers, who are working with the CFD tool. This purpose is found in the composition of the present work, where different details and aspects of the development process of SAM-Lattice are explained. The outline of the thesis will be shortly presented in this light. Chapter 2 introduces the kinetic theory, which provides the Boltzmann equation and the basics for the Lattice Boltzmann Method, which is addressed in Chapter 3. LBM is presented as a discretization of the Boltzmann equation and the assumptions and restrictions associated to this process are highlighted. In this way, the limitations of the method, like the small Mach number limit, are assignable and expandabilities are identified. Subsequently, the consistent large eddy approach for Lattice Boltzmann Methods is proposed in Chapter 4 on a detailed theoretical basis, again highlighting assumptions and restrictions. Chapter 5 gives understanding of some implementation aspects of LBM in SAM-Lattice. Verification of the CFD package by means of analytical solutions of the Navier-Stokes equations is provided in Chapter 6. The following chapter, no. 7, is dedicated to the application and validation of LBM for pump intake flows. Afterwards the thesis is closed by conclusions and suggestions for future work.

### 1.3 Physical Description of Fluid Flows

From a physical point of view, there are three different ways to describe fluid flows, depending on different observation scales. Each approach results in different mathematical equations and has its prospects and limitations.

The first approach based on the smallest molecular scales is the *microscopic approach*. In the so called molecular dynamic simulations, the dynamics of every molecule in the fluid are tracked. The motion of the molecules can be described by Newtonian mechanics. Macroscopic fluid properties are derived in molecular dynamic simulations by averaging of molecular quanti-

ties, for details see, e.g., [10]. In principle, the microscopic description of flows is applicable to all flows, but has radical practical and technical limitations: Flows of technical interest contain far too many molecules to track them all. Since one mole of a fluid already contains  $6.0225 \cdot 10^{23}$  particles, it is impossible for today's and upcoming super computers to compute technical flows in this fashion. Another fact is that fluid dynamic engineers are normally not interested in such detailed molecular information. The macroscopic flow quantities are adequate to characterize a flow for engineering applications.

The *mesoscopic approach* is based on the molecular level as well. Instead of tracking every single molecule, a statistical description is used. In *kinetic theory*, a probability distribution function is defined, which expresses the probability to find a molecule with a certain velocity at a certain position in space. From the probability distribution function all macroscopic flow quantities can be calculated. The probability distribution function is determined from a kinetic equation, which describes the conservation of the probability distribution function. More details about this equation, called Boltzmann equation, will be given in the next chapter. Kinetic theory is in general valid from continuum to free molecular flows.

The third way is the *macroscopic approach*, which is based on a continuum description of fluid flow. The macroscopic quantities are calculated from conservation equations according to the continuum theory. For example, an incompressible isothermal flow is characterized by the pressure and the velocity field. These quantities are calculated from the continuity and the Navier Stokes equations (see, e.g., [34] for more details). Macroscopic based schemes are nowadays most widely used for Computational Fluid Dynamics. But, the macroscopic approach is only applicable as long as the continuum theory is valid. The validity of the continuum theory is identified by the Knudsen number  $Kn$ , which is defined as the ratio of mean free path of molecules  $\lambda$  to a characteristic length  $L$  of the flow. The macroscopic description is valid for  $Kn \ll 1$ .

## 1.4 Mathematical Notation

For a clear understanding of the mathematical symbols and equations in this thesis, the notation in use will be shortly announced in this section. If not otherwise stated, vectors are indicated by bold lower-case symbols, e.g.,  $\mathbf{a}$  and matrices or higher order tensor, respectively, by bold upper-case symbols, like  $\mathbf{A}$ . In some cases, where it makes the situation clearer, index notation of vectors or tensors, respectively, is applied. As common in literature, the index notation is used in combination with Einstein's summation convention. The applied definitions of vector and tensor operations conform with the standard definitions for Euclidean space, see Table 1.1.

More noteworthy is the description of the Lattice Boltzmann Method. In literature, commonly all lattice quantities are non-dimensional by the grid spacing  $\Delta x$  and the time step  $\Delta t$ , what leads to a spacing  $\Delta x^l = 1$  and a time step  $\Delta t^l = 1$  in lattice units. Since these quantities are unity they are often omitted in equations in lattice units. This oftentimes complicates the un-

Table 1.1: Mathematical Definitions

<b>Operation</b>	<b>Definition</b>
Dot product	$c = \mathbf{a} \cdot \mathbf{b} = a_\alpha b_\alpha$
Dyadic product	$\mathbf{C} = \mathbf{a}\mathbf{b} = a_\alpha b_\beta$
Matrix product	$\mathbf{C} = \mathbf{A}\mathbf{B} = A_{\alpha,\beta} B_{\beta,\gamma}$
Tensor contraction	$c = \mathbf{A} : \mathbf{B} = A_{\alpha,\beta} B_{\alpha,\beta}$
Gradient of a scalar field	$\nabla g = \frac{\partial g}{\partial x_\alpha}$
Gradient of a vector field	$\mathbf{J} = \nabla \mathbf{g} = \frac{\partial g_\alpha}{\partial x_\beta}$
Divergence of a vector field	$\nabla \cdot \mathbf{g} = \frac{\partial g_\alpha}{\partial x_\alpha}$

derstanding and physical interpretation of formulae. In this thesis all quantities and equations of LBM are used in a dimensional description, to circumvent these difficulties and allow an unique interpretation.





## CHAPTER 2

---

### Kinetic Theory

---

The kinetic theory of gases describes a gas on a microscopic or molecular level. As already said in the previous Section 1.3, the condition of the gas could be completely characterized by tracking the position and the velocity of every molecule. Due to the unthinkable large number of molecules in a real gas, this is not feasible and a statistical approach must be used. The introduction to the essential concept given in this section is mainly based on the textbooks by Bird [10] and Hänel [50].

### 2.1 The Distribution Function

The position of a molecule in physical space can be characterized by its  $x$ ,  $y$  and  $z$  coordinate, which can be combined to a position vector  $\mathbf{x}$ , see Fig. 2.1(a). A volume element in physical space can be defined by  $dV = dx \cdot dy \cdot dz$ . Analog to the physical space, a velocity space can be defined, cf. Fig. 2.1(b). The velocity  $\boldsymbol{\xi}$  of a molecule is a point in the velocity space with the components  $u$ ,  $v$  and  $w$ . In velocity space a volume element can be expressed by  $d\xi = du \cdot dv \cdot dw$ . The physical and velocity space are combined to a new six dimensional space called phase space. For the statistical description of the gas a probability distribution function is now defined by:

$$f(\mathbf{x}, \boldsymbol{\xi}, t) = \frac{dN}{dV \cdot d\xi}. \quad (2.1)$$

The distribution function can be interpreted in two ways:

- Deterministically, as the number of particles  $dN$  per phase space volume  $dV \cdot d\xi$  at a position  $\mathbf{x}$  with a velocity  $\boldsymbol{\xi}$  at a given time  $t$ .

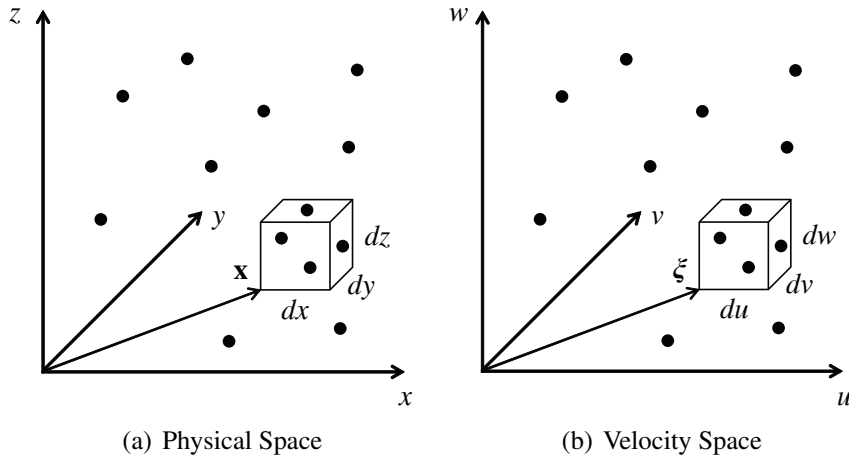


Figure 2.1: Phase Space

- Statistically, as the possibility to find a molecule in the phase space volume  $dV \cdot d\xi$  at a position  $\mathbf{x}$  with a velocity  $\xi$  at a given time  $t$ .

The distribution function is the central quantity in the kinetic theory of gases, from which all macroscopic quantities can be calculated. This is done by averaging of molecular quantities and is called establishing of moments. A moment  $\mathbf{M}$  of the distribution function is the integral over the velocity space of a function  $\varphi(\xi)$ , which is either a constant or depending on the molecular velocity  $\xi$ , multiplied with the distribution function  $f$ :

$$\mathbf{M}(\mathbf{x}, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi(\xi) \cdot f(\mathbf{x}, \xi, t) du dv dw := \int_{\xi} \varphi(\xi) \cdot f(\mathbf{x}, \xi, t) d\xi. \quad (2.2)$$

The definition of the function  $\varphi(\xi)$  is depending on the macroscopic quantity to determine. Referring to the power  $n$  of the molecular velocity  $\xi$  in the function  $\varphi$ , the corresponding moment is called a  $n$ -th order moment of the distribution function, e.g., the particle density in Eq. (2.3) is a zeroth order moment. The particle density is established for  $\varphi = 1$ , what results in:

$$n(\mathbf{x}, t) = \lim_{dV \rightarrow 0} \frac{dN}{dV} = \int_{\xi} f(\mathbf{x}, \xi, t) d\xi. \quad (2.3)$$

The macroscopic variables, which describe a flow in general are density, pressure, momentum, and temperature. These quantities are recovered by the following moments [50]:

$$\rho = m \cdot \int_{\xi} f(\mathbf{x}, \xi, t) d\xi \quad (2.4)$$

$$p = \frac{m}{3} \cdot \int_{\xi} (\xi - \mathbf{u})^2 \cdot f(\mathbf{x}, \xi, t) d\xi \quad (2.5)$$

$$\rho \cdot \mathbf{u} = m \cdot \int_{\xi} \xi \cdot f(\mathbf{x}, \xi, t) d\xi \quad (2.6)$$

$$\rho \cdot e = \frac{3}{2} \rho \cdot R \cdot T = \frac{m}{2} \cdot \int_{\xi} (\xi - \mathbf{u})^2 \cdot f(\mathbf{x}, \xi, t) d\xi, \quad (2.7)$$

$$(2.8)$$

where  $m$  is the molecular mass. Density, pressure, and temperature in the above equations are linked by the equation of state for an ideal gas. Additionally, the macroscopic system of equations, i.e., Navier-Stokes and energy equation, include the momentum flux tensor and the heat flux vector, which are expressible by moments, too:

$$\mathbf{\Pi} = m \cdot \int_{\xi} \xi \xi \cdot f(\mathbf{x}, \xi, t) d\xi \quad (2.9)$$

$$\mathbf{q} = \frac{m}{2} \cdot \int_{\xi} |\xi - \mathbf{u}|^2 \cdot (\xi - \mathbf{u}) \cdot f(\mathbf{x}, \xi, t) d\xi. \quad (2.10)$$

For the definition of the remaining, physically interpretable moments the reader is referred to [10, 50].

## 2.2 The Boltzmann Equation

As shown in the previous section, the knowledge of the distribution function permits the calculation of all macroscopic flow quantities. But, up to now, it is not clear how the distribution function  $f$  itself can be determined. Based on the conservation of molecules in a system, which is equal to mass conservation for non reacting molecules, a conditional equation for  $f$ , the Boltzmann equation, can be derived.

The change of the number of molecules  $dN$  in a volume element of the phase space  $dV \cdot d\xi$  by time, must be related to the following processes (for a more detailed presentation see [10]):

- Convective transport of the molecules in and out of the considered physical space volume  $dV$  due to the molecular velocity  $\xi$ .
- Convective transport of the molecules in and out of the considered velocity space volume  $d\xi$  due to the acceleration or deceleration of external forces.

- Gain and loss of molecules in the the considered velocity space volume  $d\xi$  due to collisions of the molecules.

The mathematical formulation of the above principles leads to the Boltzmann equation (see [10] or [50] for a full length derivation):

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \boldsymbol{\xi}} = \left( \frac{\partial f}{\partial t} \right)_{coll} . \quad (2.11)$$

For a better readability the variables of  $f = f(\mathbf{x}, \boldsymbol{\xi}, t)$  were omitted in Eq. (2.11). The different terms of the Boltzmann equation can now be readily explained: The left-hand side of Eq. (2.11) is called transport or advection term and contains the temporal and the convective change of the distribution function  $f$  in physical and velocity space. The right-hand side is referred to as collision term and represents the change of the distribution function  $f$  due to molecular collisions. For the derivation of the collision term, the assumption of dilute gas is made. This means that the mean free path  $\lambda$  between molecular collisions is much greater than the mean molecular spacing  $\delta$ , which expresses the mean volume per molecule.  $\delta$  in turn is much greater than the molecular diameter  $d$ :  $\lambda \gg \delta \gg d$ . Consequences of this assumption are that intermolecular forces are negligible and collisions of three or more molecules are very unlikely, so all collisions are regarded as binary elastic collisions between two molecules. As another consequence, the collisions can be considered instantaneous, which means the distribution function does not change during the collision.

The collision term introduced by Boltzmann is presented, for the sake of completeness, in Eq. (2.12). This collision integral is quite complicated and turns the Boltzmann equation into a mathematically difficult to solve integro-differential equation. Equation (2.12) shows the change of the distribution function  $f = f(\mathbf{x}, \boldsymbol{\xi}, t)$  due to collisions of molecules with speed  $\boldsymbol{\xi}$  and molecules with a different speed  $\boldsymbol{\xi}_1$ , which are considered in  $f_1 = f_1(\mathbf{x}, \boldsymbol{\xi}_1, t)$ . We see that the change of  $f$  depends on the pre-collision and post-collision (denoted with ') distributions and the relative speed  $\boldsymbol{\xi}_r = \boldsymbol{\xi}_1 - \boldsymbol{\xi}$  of the molecules. The collision integral has to be evaluated for all velocities  $\boldsymbol{\xi}_1 \in ] - \infty, \infty[$  and the complete differential cross-section  $A_c$ . The reader is referred to Bird [10] and Hänel [50] for further explanations of Boltzmann's collision term.

$$\left( \frac{\partial f}{\partial t} \right)_{coll} = \int_{\boldsymbol{\xi}_1} \int_{A_c} (f' \cdot f'_1 - f \cdot f_1) \cdot |\boldsymbol{\xi}_r| dA_c d\xi_1 \quad (2.12)$$

According to classical mechanics, the macroscopic properties mass, momentum, and energy are conserved in elastic collisions. Thus, these quantities are called collisional invariants.

## 2.3 The Equilibrium Distribution Function

For a physically correct behavior, the Boltzmann equation must satisfy the second law of thermodynamics. To prove this, Boltzmann defined the H-function and pointed out its relation to

the entropy  $S$  [50]:

$$H = \int_{\xi} f \ln(f) d\xi = -\frac{1}{kV} \Delta S . \quad (2.13)$$

He could show from the Boltzmann equation for a homogeneous, closed system that

$$\frac{\partial H}{\partial t} \leq 0 \Leftrightarrow \frac{\partial S}{\partial t} \geq 0 . \quad (2.14)$$

This proves the accordance of the Boltzmann equation with the second law of thermodynamics and is known as H-theorem [10, 50].

A quintessence of the H-theorem is that any distribution function  $f$  in a homogeneous, closed system will tend with time to a certain distribution function, the equilibrium or Maxwellian distribution. During this process, which is caused by collisions of molecules, the entropy is monotonically increasing up to a finite upper bound, when thermodynamic equilibrium is reached [10, 50].

The distribution function in equilibrium state is mathematically expressed by:

$$f^{eq} = \frac{n}{(2 \cdot \pi \cdot R \cdot T)^{\frac{3}{2}}} \cdot \exp\left(\frac{-(\xi - \mathbf{u})^2}{2 \cdot R \cdot T}\right) . \quad (2.15)$$

Since the transition to the equilibrium state is generated by collisions, the equilibrium distribution has the same collisional invariants than the primary distribution  $f$ .

$$\rho = m \cdot \int_{\xi} f(\mathbf{x}, \xi, t) d\xi = m \cdot \int_{\xi} f^{eq}(\mathbf{x}, \xi, t) d\xi \quad (2.16)$$

$$\rho \cdot \mathbf{u} = m \cdot \int_{\xi} \xi \cdot f(\mathbf{x}, \xi, t) d\xi = m \cdot \int_{\xi} \xi \cdot f^{eq}(\mathbf{x}, \xi, t) d\xi \quad (2.17)$$

$$\rho \cdot e = \frac{m}{2} \cdot \int_{\xi} (\xi - \mathbf{u})^2 \cdot f(\mathbf{x}, \xi, t) d\xi = \frac{m}{2} \cdot \int_{\xi} (\xi - \mathbf{u})^2 \cdot f^{eq}(\mathbf{x}, \xi, t) d\xi \quad (2.18)$$

## 2.4 BGK Approximation

The difficulties in the solution of the Boltzmann equation (Eq. (2.11)) are mainly caused by the collision term (Eq. (2.12)). The approximation of the Boltzmann equation after Bhatnagar, Gross and Krook (BGK) [9] uses a simplified collision model. The BGK collision term is defined in the following way:

$$\left(\frac{\partial f}{\partial t}\right)_{coll} = -\omega \cdot (f - f^{eq}) = -\frac{1}{\tau} \cdot (f - f^{eq}) . \quad (2.19)$$

The collision process in the BGK approximation is replaced by a relaxation of the distribution function  $f$  to the local equilibrium  $f^{eq}$ . The relaxation parameter  $\omega$  is known as collision frequency and defines the time range in which the relaxation occurs. Besides the collision frequency, its reciprocal value, the collision time  $\tau$ , is often used in literature. Physically seen, this model expresses the consequences of the H-theorem and preserves the collisional invariants.

## 2.5 The Chapman-Enskog Expansion

By means of the Chapman-Enskog expansion, continuum equations and macroscopic transport coefficients can be derived from the Boltzmann equation. Since this derivation is a tedious mathematical action, the procedure will only be roughly described for the BGK approximation here, according to the approach of [26]. Furthermore, the description is restricted to isothermal flows. Additional presentations of the Chapman-Enskog expansion can be found in [50, 116, 132].

The Chapman-Enskog expansion is a multiscale expansion of the distribution function  $f$  and the time  $t$ , for small departures from equilibrium, what correlates to  $Kn \ll 1$ , i.e., continuum limit. The distribution function  $f$  is expanded as a power series in  $\epsilon$  around local equilibrium, where the parameter  $\epsilon$  can be interpreted as the Knudsen number (see also [26, 116, 132]):

$$f = f^{eq} + \epsilon f^1 + \epsilon^2 f^2 + \epsilon^3 f^3 + \dots \quad (2.20)$$

Analogically, the time  $t$  derivative is expanded as power series in  $\epsilon$  as well:

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial t_0} + \epsilon \frac{\partial}{\partial t_1} + \dots \quad (2.21)$$

The timescales in Eq. (2.21) can be interpreted as advective ( $t_0$ ) and viscous ( $t_1$ ) timescales, according to departure from equilibrium. The next ingredient is to introduce the parameter  $\epsilon$  in the collision time  $\tau = \epsilon \tilde{\tau}$ . For convenience, external forces will be neglected in the following. The BGK approximation of the Boltzmann equation reads then as:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f}{\partial \mathbf{x}} = -\frac{1}{\epsilon \tilde{\tau}} (f - f^{eq}). \quad (2.22)$$

Inserting Eqs. (2.20) and (2.21) in Eq. (2.22) and sorting on the basis of  $\epsilon$ , brings up for  $O(1)$  and  $O(\epsilon)$ :

$$\frac{\partial f^{eq}}{\partial t_0} + \boldsymbol{\xi} \cdot \frac{\partial f^{eq}}{\partial \mathbf{x}} = -\frac{1}{\tilde{\tau}} f^1 \quad (2.23)$$

$$\epsilon \left( \frac{\partial f^{eq}}{\partial t_1} + \frac{\partial f^1}{\partial t_0} + \boldsymbol{\xi} \cdot \frac{\partial f^1}{\partial \mathbf{x}} \right) = -\epsilon \frac{1}{\tilde{\tau}} f^2. \quad (2.24)$$

The zeroth and first order moments of  $f^n$ ,  $n \geq 1$ , are 0 due to collisional invariance. Equation (2.23) can be interpreted as a zeroth order truncation of the  $\epsilon$  expansion. Establishing the zeroth and first order moments of this equation yields the Euler equations:

$$\frac{\partial \rho}{\partial t_0} + \frac{\partial \rho \mathbf{u}}{\partial \mathbf{x}} = 0 \quad (2.25)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t_0} + \frac{\partial \Pi^{eq}}{\partial \mathbf{x}} = 0 \quad (2.26)$$

$$\text{with: } \mathbf{\Pi}^{eq} = m \cdot \int_{\xi} \xi \xi \cdot f^{eq} d\xi = p\mathbf{I} + \rho\mathbf{u}\mathbf{u}.$$

The zeroth and first order moments of Eq. (2.24) produce:

$$\frac{\partial \rho}{\partial t_1} = 0 \quad (2.27)$$

$$\epsilon \left( \frac{\partial \rho \mathbf{u}}{\partial t_1} + \frac{\partial \mathbf{\Pi}^1}{\partial \mathbf{x}} \right) = 0. \quad (2.28)$$

By combination of Eq. (2.25), (2.27), and Eq. (2.26), (2.28), and neglecting terms of  $O(\epsilon^2)$ , the Navier-Stokes equations are recovered. This correlates to a truncation of the power series at first order in  $\epsilon$ .

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho \mathbf{u}}{\partial \mathbf{x}} = 0 \quad (2.29)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \frac{\partial (\mathbf{\Pi}^{eq} + \epsilon \mathbf{\Pi}^1)}{\partial \mathbf{x}} = 0 \quad (2.30)$$

The difference between the Euler momentum equations (2.26) and the Navier-Stokes momentum equations (2.30) is the friction term, which is expressed by the stress tensor  $\boldsymbol{\sigma} = -\epsilon \mathbf{\Pi}^1$ . A relation between  $\mathbf{\Pi}^1$  and the equilibrium  $f^{eq}$  can be achieved by building the second order moment of Eq. (2.23):

$$\frac{\partial \mathbf{\Pi}^{eq}}{\partial t_0} + \frac{\partial \int_{\xi} \xi \xi \xi f^{eq} d\xi}{\partial \mathbf{x}} = -\frac{1}{\tilde{\tau}} \mathbf{\Pi}^1. \quad (2.31)$$

The terms on the left hand side of Eq. (2.31) can be expressed in terms of macroscopic quantities [26]:

$$\frac{\partial \Pi_{\alpha\beta}^{eq}}{\partial t_0} = \frac{\partial}{\partial t_0} (RT\rho\delta_{\alpha\beta} + \rho u_{\alpha}u_{\beta}) \quad (2.32)$$

$$\int_{\xi} \xi_{\alpha}\xi_{\beta}\xi_{\gamma} f^{eq} d\xi = \rho u_{\alpha}u_{\beta}u_{\gamma} + RT\rho (u_{\alpha}\delta_{\beta\gamma} + u_{\beta}\delta_{\gamma\alpha} + u_{\gamma}\delta_{\alpha\beta}). \quad (2.33)$$

For incompressible, isothermal flows Eq. (2.31) reduces to [26, 50]:

$$\Pi_{\alpha\beta}^1 = -\tilde{\tau}RT\rho \left( \frac{\partial u_{\beta}}{\partial x_{\alpha}} + \frac{\partial u_{\alpha}}{\partial x_{\beta}} \right). \quad (2.34)$$

With the continuum definition of the stress tensor [106]

$$\sigma_{\alpha\beta} = \mu \left( \frac{\partial u_{\beta}}{\partial x_{\alpha}} + \frac{\partial u_{\alpha}}{\partial x_{\beta}} \right) \quad (2.35)$$

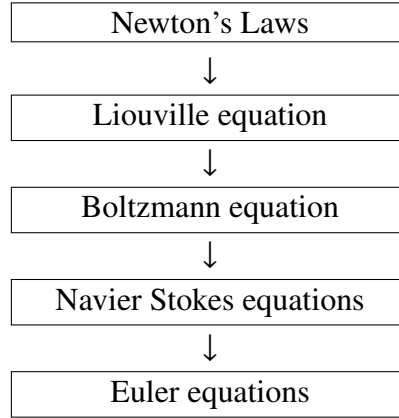


Figure 2.2: BBGKY Hierarchy

and Eq. (2.34), a relationship between the collision time  $\tau$  and the viscosity of the fluid, which is the macroscopic transport coefficient of momentum, can be established. Using the relation  $\sigma_{\alpha\beta} = -\epsilon\Pi_{\alpha\beta}^1$  one gets:

$$\mu = \epsilon\tilde{\tau}\rho RT = \tau\rho RT = \frac{\rho RT}{\omega} \quad (2.36)$$

$$\nu = \frac{\mu}{\rho} = \tau RT = \frac{RT}{\omega} . \quad (2.37)$$

As shown above, the zeroth order truncation of the  $\epsilon$  expansion results in the Euler equations and the first order truncation in the Navier-Stokes equations. Higher order truncations lead to Burnett and super Burnett equations. The derivation of the continuum equations from kinetic equations is part of the BBGKY (Bogoliubov-Born-Green-Kirkwood-Yvon) hierarchy. This hierarchy expresses the bottom up approach of describing fluid flows, visualized in Fig. 2.2.

Starting at microscopic level, the motion can be described by Newton's Laws. Applying kinetic theory, the Liouville equation can be established, which will not be explained here (see [10] for details). The Liouville equation can, under certain assumptions, be transferred to the Boltzmann equation, which itself leads to the Navier-Stokes and Euler equations by the discussed Chapman-Enskog expansion.



---

## The Lattice Boltzmann Method

---

Historically seen, the Lattice Boltzmann Method emerged from a numerical scheme called Lattice Gas Automata as a solver for the Navier-Stokes equations. But, LBM can also be directly derived from the Boltzmann equation. Thus, the Lattice Boltzmann Method is a numerical technique for solving the Boltzmann equation under certain approximations. This derivation and further essential topics of LBM will be shown in the following section. For a detailed illustration of Lattice Gas Automata and the historical development of LBM the reader is referred to the textbooks of Succi [116] and Wolf-Gladrow [132].

In the context of Lattice Boltzmann Methods, the commonly used distribution is slightly different to the definition in Eq. (2.1):

$$f(\mathbf{x}, \boldsymbol{\xi}, t) = m \cdot \frac{dN}{dV \cdot d\xi} \quad (3.1)$$

The distribution function here includes the molecular mass  $m$  compared to Eq. (2.1). This form of distribution is sometimes called density distribution function and will be used in the following.

### 3.1 The Lattice Boltzmann Equation

Every numerical scheme requires a discretization of the continuous equation and the computational domain. Based on the continuous BGK approximation of the Boltzmann equation (Eqs. (2.11) and (2.19)), a discrete equation, the Lattice Boltzmann equation, is now established.

### 3.1.1 Discretization of Velocity Space

Velocity space can be discretized by several mathematical methods. Here, the so called discrete velocity method ([1, 10, 110]) is used, since it is intuitive and illustrative. In this method the molecules are restricted to have only a small number of possible velocities, which are discrete values of the velocity space. The calculation of moments according to Eq. (2.2) by integrating over the continuous velocity space must then be converted into a discrete evaluation by a Gauss-Hermite quadrature [52].

$$\mathbf{M}(\mathbf{x}, t) = \int_{-\infty}^{\infty} \varphi(\boldsymbol{\xi}) f(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} = \sum_{i=0}^{q-1} \varphi(\boldsymbol{\xi}_i) f_i \quad (3.2)$$

$$f_i = \tilde{w}_i f(\mathbf{x}, \boldsymbol{\xi}_i, t) \quad (3.3)$$

The factors  $\tilde{w}_i$  and  $\boldsymbol{\xi}_i$  are the weights and nodes or abscissae (i.e. discrete velocities) of the quadrature. The parameter  $q$  is the number of discrete velocities. Possible arrangements of discrete velocities, or quadrature points, respectively, will be discussed in the next section. The identity in Eq. (3.2) implies that the distribution  $f$  can be expressed by a polynomial in  $\boldsymbol{\xi}$ . This can indeed be done by an expansion with a Hermite polynomial, see [110, 111] for details.

The equilibrium distribution function needs discretization as well, to establish their moments in discrete velocity space. The application of the quadrature rules requires the expansion of  $f^{eq}$  as a polynomial. This can be done by a Taylor series in  $\mathbf{u}$  of Eq. (2.15), which is commonly truncated at second order. (Remember that  $f$  is multiplied by the molecular mass in the LBM context.)

$$f^{eq} = \frac{\rho}{(2\pi RT)^{\frac{3}{2}}} \exp\left(\frac{-\boldsymbol{\xi}^2}{2RT}\right) \left\{ 1 + \frac{(\boldsymbol{\xi} \cdot \mathbf{u})}{RT} + \frac{(\boldsymbol{\xi} \cdot \mathbf{u})^2}{2(RT)^2} - \frac{\mathbf{u}^2}{2RT} \right\} + O(u^3)$$

$$f^{eq} = w(\boldsymbol{\xi}) \rho \left\{ 1 + \frac{(\boldsymbol{\xi} \cdot \mathbf{u})}{RT} + \frac{(\boldsymbol{\xi} \cdot \mathbf{u})^2}{2(RT)^2} - \frac{\mathbf{u}^2}{2RT} \right\} + O(u^3) \quad (3.4)$$

The coefficients before the curly brackets in the first presentation of the truncated equilibrium are combined in the weight function  $w(\boldsymbol{\xi})$  in Eq. (3.4). Equation (3.4) can also be seen as a Taylor expansion of the equilibrium in the Mach number (Ma) :

$$Ma = \frac{|\mathbf{u}|}{c_s} \quad (3.5)$$

$$c_s = \sqrt{\left. \frac{dp}{d\rho} \right|_T} = \sqrt{RT} \quad (3.6)$$

$$\frac{p}{\rho} = RT. \quad (3.7)$$

The Mach number is the ratio of the magnitude of the fluid velocity  $|\mathbf{u}|$  to the speed of sound  $c_s$ . The isothermal speed of sound  $c_s$  for an ideal gas is given by Eq. (3.6) and the equation of state for an ideal gas is specified in Eq. (3.7). The truncation of Eq. (3.4) at  $\mathbf{u}^2$  corresponds to a truncation at  $Ma^2$  and is therefore only valid for small Mach numbers. Fluid flows are generally considered incompressible for small Mach numbers, i.e.,  $Ma < 0.3$ . This approximation is thus adequate for the restriction made to incompressible, isothermal flows in this thesis.

Moments of the truncated equilibrium distribution, Eq. (3.4), for the discrete velocities  $\xi_i$  result from Eq. (3.2), where  $f_i = \tilde{w}_i f_i^{eq} = \tilde{w}_i f^{eq}(\rho, \mathbf{u}, \xi_i)$  in this case. Commonly, the discrete value of the weight function  $w(\xi_i)$  (Eq. (3.4)) and the value of quadrature weight  $\tilde{w}_i$  (Eq. 3.3) are combined to a single weight factor  $w_i$ , as done in Eq. (3.8).

$$f_i^{eq} = \tilde{w}_i w(\xi_i) \rho \left\{ 1 + \frac{(\xi_i \cdot \mathbf{u})}{c_s^2} + \frac{(\xi_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right\}$$

$$f_i^{eq} = w_i \rho \left\{ 1 + \frac{(\xi_i \cdot \mathbf{u})}{c_s^2} + \frac{(\xi_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right\} \quad (3.8)$$

As we can see, the space and time dependence of  $f_i^{eq}$  is only indirect through the space and time dependence of  $\rho(\mathbf{x}, t)$  and  $\mathbf{u}(\mathbf{x}, t)$ .

Since the macroscopic quantities are established from the discrete formulae, the distribution function needs only to be evolved in space and time for the discrete velocities  $\xi_i$ . In the absence of external forces, the resulting *discrete velocity Boltzmann-BGK equation*, hereafter called discrete Boltzmann equation, is shown in Eq. (3.9) [1, 110]. The treatment of external forces will be concerned later. Equation (3.9) is a set of  $q$  nonlinear first order differential equations for the discrete values of the distribution function.

$$\frac{\partial f_i}{\partial t} + \xi_i \cdot \frac{\partial f_i}{\partial \mathbf{x}} = -\frac{1}{\tau} \cdot (f_i - f_i^{eq}) \quad (i = 0, \dots, q-1) \quad (3.9)$$

For recovering the isothermal Navier-Stokes equations in the continuum limit, the moments up to second order, i.e., density, momentum, and stress tensor, must be correctly recovered by the discrete model. By applying a Chapman-Enskog expansion to the discrete Boltzmann equation with the discrete equilibrium distribution presented here, it is shown that this model approximates the Navier-Stokes equations with second order accuracy in the Knudsen number  $Kn$  [1].

Another, mathematically very formal way to derive the discrete Boltzmann equation is the expansion of the distribution function with Hermite polynomials. Shan and He [110] show that this approach is equivalent to the discrete velocity method used here. Nevertheless, the expansion by Hermite polynomials yields a direct relation between the truncation order of the polynomial and the representation of the physical system. Equation (3.4) can be interpreted as a second order truncated Hermite polynomial, see [110, 111]. By truncating the Hermite polynomial at higher orders, LBM schemes for simulating thermal and higher Mach number flows can be developed. Higher order truncation means that higher order terms of fluid velocity

or Mach number, respectively, will occur in the discrete equilibrium distribution function and a higher number of Gaussian quadrature points  $\xi_i$  are needed [111].

### 3.1.2 Discretization in Space and Time

After discretization in velocity space, the discretization in physical space and time is now performed to obtain a model, ready for numerical simulation. The discrete Boltzmann equation can be rewritten with a total or Lagrangian derivative as:

$$\frac{\partial f_i}{\partial t} + \xi_i \cdot \frac{\partial f_i}{\partial \mathbf{x}} = \frac{Df}{Dt} = -\frac{1}{\tau} \cdot (f_i - f_i^{eq}). \quad (3.10)$$

By introducing the convective scaling  $\Delta \mathbf{x} = \xi_i \Delta t$ , Eq. (3.10) can be integrated along a characteristic for a time interval [26, 54]:

$$\int_0^{\Delta t} \frac{Df_i(\mathbf{x} + \xi_i s, t + s)}{Ds} ds = \int_0^{\Delta t} -\frac{1}{\tau} [f_i(\mathbf{x} + \xi_i s, t + s) - f_i^{eq}(\mathbf{x} + \xi_i s, t + s)] ds. \quad (3.11)$$

While the total derivative on the left-hand side of Eq. (3.11) integrates directly, the right-hand side needs approximation. Since the discrete Boltzmann equation is second order accurate in terms of  $Kn$ , it is reasonable to preserve the order and use a second order accurate quadrature rule. Here the trapezoidal rule is applied:

$$f_i(\mathbf{x} + \xi_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = -\frac{\Delta t}{2\tau} \left[ f_i(\mathbf{x} + \xi_i \Delta t, t + \Delta t) - f_i^{eq}(\mathbf{x} + \xi_i \Delta t, t + \Delta t) \right. \\ \left. + f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t) \right] + \mathcal{O}(\Delta t^2). \quad (3.12)$$

The values of the equilibrium  $f_i^{eq}(\mathbf{x} + \xi_i \Delta t, t + \Delta t)$  are not known a priori and depend on  $f_i(\mathbf{x} + \xi_i \Delta t, t + \Delta t)$ ,  $i \in \{0, \dots, q-1\}$ . This creates an implicit system of nonlinear, algebraic equations from Eq. (3.12). By the following substitution, the system of equations is transformed to a fully explicit system [26, 51, 54]:

$$\bar{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) + \frac{\Delta t}{2\tau} \left[ f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t) \right]. \quad (3.13)$$

After some algebra one obtains from Eq. (3.12) with (3.13) this explicit set of equations [26, 51, 54]:

$$\bar{f}_i(\mathbf{x} + \xi_i \Delta t, t + \Delta t) - \bar{f}_i(\mathbf{x}, t) = -\frac{\Delta t}{\tau + 0.5\Delta t} \left[ \bar{f}_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t) \right]. \quad (3.14)$$

The fraction in Eq. (3.14) is from a mathematical point of view a relaxation parameter, which describes relaxation to equilibrium. In analogy to the BGK model, this parameter is considered as a dimensionless collision time, denoted by  $\Omega$ . Compared to the continuous model, the additional parameter  $0.5\Delta t$  occurs in the relaxation parameter of the discrete model. This numerical

viscosity, sometimes called discrete lattice effect [47], is absorbed into the physical model, i.e., the relaxation parameter, to obtain correct macroscopic transport coefficients [51]. By using the relationships (2.37) and (3.6),  $\Omega$  becomes a function of the kinematic fluid viscosity:

$$\Omega = \frac{\Delta t}{\bar{\tau}} = \frac{\Delta t}{\tau + 0.5\Delta t} = \frac{c_s^2 \Delta t}{\nu + 0.5c_s^2 \Delta t}. \quad (3.15)$$

Finally the *Lattice Boltzmann equation* (LBE), sometimes called Lattice BGK equation, is achieved from Eq. (3.16) with the dimensionless collision time. (Note: From Eq. (3.13) it follows  $f_i^{eq} = \bar{f}_i^{eq}$ .)

$$\bar{f}_i(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) = \bar{f}_i(\mathbf{x}, t) + \Omega \left[ \bar{f}_i^{eq}(\mathbf{x}, t) - \bar{f}_i(\mathbf{x}, t) \right] \quad (3.16)$$

The Lattice Boltzmann equation is a set of discrete equations for the evolution of the discrete values of the distribution function  $\bar{f}_i$ . The next step is to evaluate the consequences of the substitution process on the macroscopic or hydrodynamic variables. Since  $f_i^{eq} = \bar{f}_i^{eq}$ , i.e., the collision operator preserves mass and momentum, the zeroth and first order moments remain unchanged:

$$\rho = \sum_{i=0}^{q-1} \bar{f}_i = \sum_{i=0}^{q-1} \bar{f}_i^{eq} = \sum_{i=0}^{q-1} f_i^{eq} = \sum_{i=0}^{q-1} f_i \quad (3.17)$$

$$\rho \mathbf{u} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \bar{f}_i = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \bar{f}_i^{eq} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{eq} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i. \quad (3.18)$$

For the momentum flux tensor things are bit different:

$$\begin{aligned} \bar{\Pi} &= \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i \bar{f}_i \\ &= \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i f_i + \frac{\Delta t}{2\tau} \left[ \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i f_i - \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i f_i^{eq} \right] \\ &= \bar{\Pi} + \frac{\Delta t}{2\tau} [\bar{\Pi} - \bar{\Pi}^{eq}] \\ \text{with : } \bar{\Pi}^{neq} &= \bar{\Pi} - \bar{\Pi}^{eq} \text{ and } \bar{\Pi}^{eq} = \bar{\Pi}^{eq} \\ \Rightarrow \bar{\Pi}^{neq} &= \left( 1 + \frac{\Delta t}{2\tau} \right)^{-1} [\bar{\Pi} - \bar{\Pi}^{eq}] \\ \bar{\Pi}^{neq} &= \left( 1 + \frac{\Delta t}{2\tau} \right)^{-1} \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i (\bar{f}_i - \bar{f}_i^{eq}) \\ &= \left( 1 + \frac{\Delta t}{2\tau} \right)^{-1} \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i \bar{f}_i^{neq}. \end{aligned} \quad (3.19)$$

$$\bar{\Pi}^{neq} = \left( 1 + \frac{\Delta t}{2\tau} \right)^{-1} \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i \bar{f}_i^{neq}. \quad (3.20)$$

Equation (3.20) shows the influence of the discrete lattice effect on the non equilibrium part of the momentum flux tensor. The non equilibrium part of the distribution  $f_i^{neq} = f_i - f_i^{eq}$  is an approximation for the first order term  $f_i^1$  of the Chapman-Enskog expansion (cf. Eq. (2.20)). An important result of the Chapman-Enskog expansion shown above, is the correlation between the macroscopic stress tensor of the Navier-Stokes equations  $\sigma$  and  $\Pi^1$ . Using this result and Eqs. (2.37), (3.6), and (3.20) yields an relationship for the macroscopic stress tensor and the momentum flux tensor of the discrete model:

$$\begin{aligned}
\sigma &= -\epsilon \Pi^1 \approx -\Pi^{neq} \\
\sigma &= -\left(1 + \frac{c_s^2 \Delta t}{2\nu}\right)^{-1} \sum_{i=0}^{q-1} \xi_i \xi_i (\bar{f}_i - \bar{f}_i^{eq}) \\
&= -\left(\frac{2\nu}{2\nu + c_s^2 \Delta t}\right) \sum_{i=0}^{q-1} \xi_i \xi_i (\bar{f}_i - \bar{f}_i^{eq}) \\
&= -\left(\frac{\nu}{\nu + 0.5c_s^2 \Delta t} \cdot \frac{c_s^2 \Delta t}{c_s^2 \Delta t}\right) \sum_{i=0}^{q-1} \xi_i \xi_i (\bar{f}_i - \bar{f}_i^{eq}) \\
\sigma &= -\frac{\Omega \nu}{c_s^2 \Delta t} \sum_{i=0}^{q-1} \xi_i \xi_i (\bar{f}_i - \bar{f}_i^{eq}) \\
&= -\left(1 - \frac{1}{2}\Omega\right) \sum_{i=0}^{q-1} \xi_i \xi_i (\bar{f}_i - \bar{f}_i^{eq}) .
\end{aligned} \tag{3.21}$$

The LBE derived here, is a second order accurate discretization of the discrete Boltzmann equation in space and time and, thus, it recovers in the continuum limit the Navier-Stokes equation with second order accuracy. In what follows, only the distribution function with an overbar will be used, thus the overbar notation will be omitted for simplicity.

The discretization procedure shown here is not the only way to discretize the discrete Boltzmann equation in space and time. As demonstrated by Chen and Doolen [21] the time derivative in Eq. (3.9) can be discretized by an Euler forward difference and the convective term by a first order upwind scheme. After introducing the convective scaling  $\Delta \mathbf{x} = \xi_i \Delta t$ , the Lattice Boltzmann equation is recovered. But, in contrast to the above derivation, the expression of the relaxation parameter  $\Omega$  is not derived during the discretization procedure. Instead, it must be determined by a subsequent Chapman-Enskog expansion of the Lattice Boltzmann equation. This procedure reproduces the second order accuracy of LBM to the Navier-Stokes equations as well [51]. Another way to derive the Navier-Stokes equations from the LBE and to show the accuracy of the scheme is the so called asymptotic analysis as proposed by Junk et al. [63]. The different links between the Boltzmann-BGK equation, the discrete Boltzmann equation, and the Navier-Stokes equations are schematically shown up in Fig. 3.1.

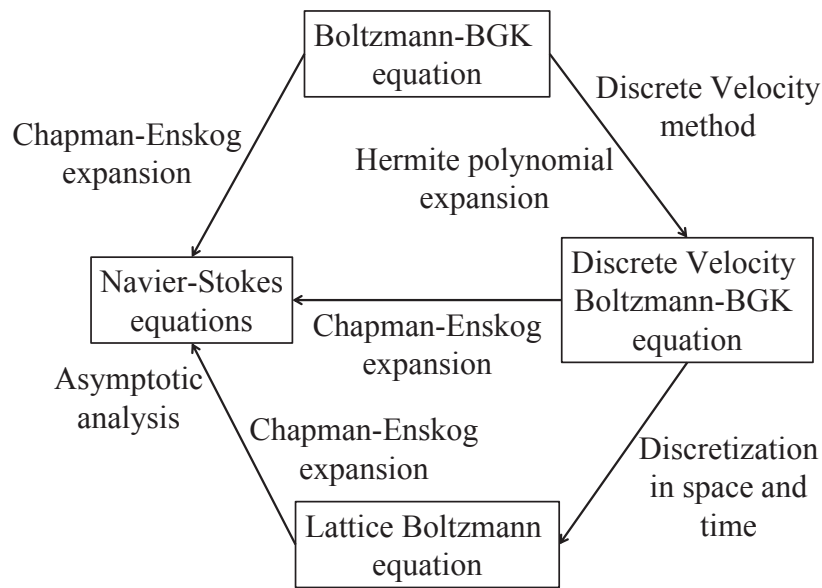


Figure 3.1: Links to Boltzmann-BGK Equation

## 3.2 Discrete Phase Space Model

In the preceding section, the Boltzmann equation has been discretized. Therefore, the discrete velocity space was derived by restricting the molecules to certain velocities, which were not clearly specified up to now. The set of discrete velocities must fulfill different requirements.

The most important criteria to meet is Eq. (3.2), which means that the moments of the discrete distribution function are identical to the ones based on the continuous velocity space. But, it can be shown that matching of all moments requires an infinite number of velocities [17]. Since the goal of the discretization is to use only a finite set of velocities, some limitations are needed. Thus, the discrete moments are restricted to be equal to their continuous counterparts up to a certain order  $m$ , depending on the physics to be represented. Chen and Shan [17] state that an accurate calculation of the discrete non-equilibrium moment at order  $m$  requires a correct evaluation of the discrete equilibrium moments at order  $n = m + 1$ . Furthermore, the accurate calculation of discrete moments defines the degree of precision  $q \geq m + n$  of the Gauss-Hermite quadrature. The precision requirements of the quadrature are equivalent to satisfying the isotropic tensor conditions of Eq. (3.24) by the discrete velocity set [89]. Meeting the isotropic conditions results in rotational invariance of the hydrodynamic moments [20, 46].

Before coming to these isotropic conditions, some definitions are needed:

$$\mathbf{E}^{(n)} = \sum_{i=0}^{q-1} w_i \underbrace{\xi_i \xi_i \dots \xi_i}_n \quad (3.22)$$

$$E_{\alpha_1, \alpha_2, \dots, \alpha_n}^{(n)} = \sum_{i=0}^{q-1} w_i \xi_{i, \alpha_1} \xi_{i, \alpha_2} \dots \xi_{i, \alpha_n} \cdot \quad (3.23)$$

The quantity  $\mathbf{E}^{(n)}$  is a tensor of  $n$ -th order build from the discrete velocities  $\xi_i$  and the weights  $w_i$  (cf. Eq. (3.8)), accordingly  $\xi_i \xi_i \dots \xi_i$  has to be understood as a dyadic product.  $E_{\alpha_1, \alpha_2, \dots, \alpha_n}^{(n)}$  can be interpreted as one component of the tensor for fixed  $\alpha_1, \alpha_2, \dots, \alpha_n$  or by the Einstein convention as the tensor itself. The index  $\alpha_j = \{x, y, z\}$  expresses a component of the  $j$ -th velocity vector in the tensor product.

$$E_{\alpha_1, \alpha_2, \dots, \alpha_n}^{(n)} = \begin{cases} (RT)^{n/2} \Delta_{\alpha_1, \alpha_2, \dots, \alpha_n}^{(n)} & n = \text{even integer}, n \leq q \\ 0 & n = \text{odd integer}, n \leq q \end{cases} \quad (3.24)$$

In the isotropic conditions (Eq. (3.24))  $\Delta_{\alpha_1, \alpha_2, \dots, \alpha_n}^{(n)}$  is the  $n$ -th order delta function defined as a summation of  $n/2$  products of Kronecker deltas, see [17] for details.

In what follows, the conditions of Eq. (3.24) will be applied to a set of 19 discrete velocities in a 3 dimensional Cartesian space, commonly named D3Q19 lattice. The notation DdQq goes back to [98], d expresses the dimension of the space and q the number of discrete velocities. The term *lattice* is used as a synonym for the discrete phase space or often interchangeable for the discrete velocity space, only. Details about other popular lattices, like D3Q15 and D3Q27 can be found in [46, 85, 98]. The D3Q19 lattice, which is used in SAM-Lattice, is numerically more stable than the D3Q15 lattice and provides a good compromise between accuracy and computational efficiency compared to D3Q15 and D3Q27 [29, 85].

In the D3Q19 lattice, the physical space is discretized by an equally spaced cubic grid. The D3Q19 lattice is shown in Fig. 3.2, including the following 19 discrete velocities:

$$\begin{aligned} \xi_i^{(0)} &= (0, 0, 0) & i &= 18 \\ \xi_i^{(1)} &= (\pm\tilde{\xi}, 0, 0), (0, \pm\tilde{\xi}, 0), (0, 0, \pm\tilde{\xi}) & i &= 0, \dots, 5 \\ \xi_i^{(2)} &= (\pm\tilde{\xi}, \pm\tilde{\xi}, 0), (\pm\tilde{\xi}, 0, \pm\tilde{\xi}), (0, \pm\tilde{\xi}, \pm\tilde{\xi}) & i &= 6, \dots, 17 \\ \tilde{\xi} &= \frac{\Delta x}{\Delta t} \end{aligned} \quad (3.25)$$

The velocity vectors originate from every computation node of the cubic grid and end up at the next neighbor in the according direction. They are categorized into three different groups, depending on the length of the vectors. This is usually called energy level. Energy level 0 contains only the zero velocity, molecules with this velocity rest at the current node. The six velocities belonging to energy level 1 point to the next neighbor nodes in Cartesian directions.



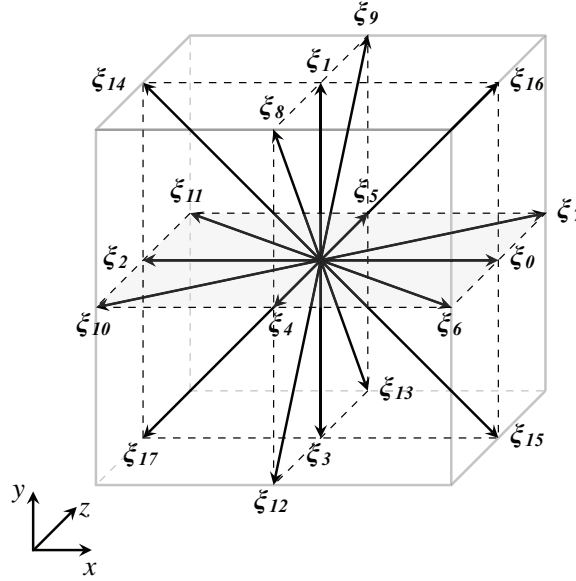


Figure 3.2: D3Q19 Lattice

The left twelve vectors in energy level 2 are chosen to point to the remaining next neighbor nodes, with exception of the space diagonal neighbors. For every discrete velocity, the D3Q19 lattice contains a velocity vector in the opposed direction. This is called parity symmetry. Due to the symmetry of the lattice and the required invariance of the hydrodynamic moments, the weights  $w_i$  can only be different for the different energy levels, i.e., all velocities in the same energy level have the same weight [20]. By the definition of the molecular velocity  $\tilde{\xi}$ , which was already used for the discretization of the Boltzmann equation in the previous section, molecules fly exactly from one node to the according neighbor during one time step.

Let us start with the demonstration of Eq. (3.24) for the D3Q19 lattice. A consequence of the parity symmetry of the D3Q19 lattice is, that the tensor of Eq. (3.24) is zero for odd integers. For even integers  $n \in \{0, 2, 4\}$ , these three conditions follow [50]:

$$E^{(0)} = \sum_{i=0}^{18} w_i = 1 \quad (3.26)$$

$$E_{\alpha_1, \alpha_2}^{(2)} = \sum_{i=0}^{18} w_i \tilde{\xi}_{i, \alpha_1} \tilde{\xi}_{i, \alpha_2} = RT \delta_{\alpha_1 \alpha_2} = c_s^2 \delta_{\alpha_1 \alpha_2} \quad (3.27)$$

$$\begin{aligned} E_{\alpha_1, \alpha_2, \alpha_3, \alpha_4}^{(4)} &= \sum_{i=0}^{18} w_i \tilde{\xi}_{i, \alpha_1} \tilde{\xi}_{i, \alpha_2} \tilde{\xi}_{i, \alpha_3} \tilde{\xi}_{i, \alpha_4} \\ &= RT^2 (\delta_{\alpha_1 \alpha_2} \delta_{\alpha_3 \alpha_4} + \delta_{\alpha_1 \alpha_3} \delta_{\alpha_2 \alpha_4} + \delta_{\alpha_1 \alpha_4} \delta_{\alpha_2 \alpha_3}) \\ &= c_s^4 (\delta_{\alpha_1 \alpha_2} \delta_{\alpha_3 \alpha_4} + \delta_{\alpha_1 \alpha_3} \delta_{\alpha_2 \alpha_4} + \delta_{\alpha_1 \alpha_4} \delta_{\alpha_2 \alpha_3}) . \end{aligned} \quad (3.28)$$

The solution of Eqs. (3.26) - (3.28) yields to the ratio of the molecular velocity to the speed of sound and to the weights for the three different energy levels:

$$\frac{\tilde{\xi}}{c_s} = \sqrt{3} \quad (3.29)$$

$$\begin{aligned} w_i &= \frac{1}{3} & i &= 18 \\ w_i &= \frac{1}{18} & i &= 0, \dots, 5 \\ w_i &= \frac{1}{36} & i &= 6, \dots, 17. \end{aligned} \quad (3.30)$$

As said before, modeling of incompressible, isothermal flows requires the correct establishing of moments up to second order, i.e.,  $m = 2$ . Consequently, the equilibrium moments up to order  $n = 3$  must be correctly evaluated in the discrete model. Nie et al. [89] show that the correct calculation of the third order equilibrium moment, which is the equilibrium heat flux vector, requires in general at least 39 velocity directions in 3D and a third order expansion of  $f_i^{eq}$ . This is not fulfilled by the D3Q19 lattice and leads to a non Galilean invariant viscous stress [28, 89]. The error in the viscous stress is of  $O(Ma^3)$  [28] and thus negligible for small  $Ma$ , i.e., incompressible flow, what is also turned out by the numerical investigation of [89]. In practice, the Galilean invariant D3Q39 lattice or models with even more velocities are hardly used, due to the considerable higher computational costs and memory demands.

With the quantities derived in this section, the discrete model is completely described. In the numerical scheme, the Lattice Boltzmann equation (3.16) is solved every time step, at every computation node for every discrete velocity. The remaining parts of this section introduce further basic ingredients of the numerical scheme.

### 3.3 Collision Operators

Besides the BGK model introduced in detail before, different other collision operators are in use for Lattice Boltzmann methods. In this section we will present the multiple relaxation time model (MRT), which is used in SAM-Lattice in detail. Other collision models, like the entropic [6] or cascaded [43] models will not be discussed here because they are currently not implemented in SAM-Lattice. The interested reader is referred to the mentioned literature.

#### 3.3.1 Single Relaxation Time Model

The term single relaxation time model (SRT) is often used as a synonym for the classical BGK model. In the BGK model, a single relaxation parameter  $\Omega$  (see Eq. (3.16)) is used for relaxation to equilibrium. Thus, all physical processes are relaxed with the same rate or time scale,

respectively, to their according equilibrium. A drawback and direct effect of this is that all physical transport coefficients, which are related to the relaxation processes, must be equal. If we consider a Lattice Boltzmann Method feasible for thermal calculations, a constant Prandtl number of unity is the consequence, since thermal conductivity and kinematic viscosity are equal [109]. But from a physical point of view the Prandtl number is not fixed to 1.

### 3.3.2 Multiple Relaxation Time Model

Physically seen, the different quantities or processes and their modes have different time scales to relax to the according equilibrium. This fact is considered in the LBM by the multiple relaxation time model, which is still based on the BGK idea of relaxation to equilibrium. The collision in MRT is performed in the moment space, instead of the velocity space in SRT. In this way, for every moment a different relaxation time can be used and thus different physical transport coefficients can be realized. With the MRT collision operator, the LBE or sometimes called generalized Lattice Boltzmann equation reads as [46]:

$$f_i(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \sum_{j=0}^{q-1} (\mathbf{M}^{-1} \hat{\mathbf{S}} \mathbf{M})_{ij} [f_j^{eq} - f_j]. \quad (3.31)$$

The transformation matrix  $\mathbf{M}$  is a linear mapping between the discrete velocity space and the discrete moment space, the diagonal matrix  $\hat{\mathbf{S}}$  contains the multiple relaxation times and the inverse matrix  $\mathbf{M}^{-1}$  maps from moment back to velocity space. In the following, a isothermal MRT model for the D3Q19 lattice is presented, based on the work of d'Humières et al. [29], who proposed this for the first time.

For a linear, invertible mapping from velocity space to moment space, 19 linear independent moments must be defined, which build the moment vector  $\mathbf{m}$ :

$$\begin{aligned} \mathbf{m} &= (\rho, e, \epsilon, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{xz}, m_x, m_y, m_z)^T \\ &= (m_0, m_1, \dots, m_{18})^T. \end{aligned} \quad (3.32)$$

Many of the moments in Eq. (3.32) have a physical interpretation, which is listed below, in order of occurrence: the fluid density ( $\rho$ ), the kinetic energy ( $e$ ), square of the kinetic energy ( $\epsilon = e^2$ ), the fluid momentum ( $j_x, j_y, j_z$ ), the energy flux ( $q_x, q_y, q_z$ ) and the momentum flux tensor ( $3p_{xx}, p_{ww}, p_{xy}, p_{yz}, p_{xz}$ ). The moments  $\pi_{xx}$  and  $\pi_{ww}$  are of quartic order and have no direct physical interpretation. The same holds for  $m_x, m_y$  and  $m_z$ , which are of cubic order in  $\boldsymbol{\xi}$ .

The quadrature for establishing the moments of the discrete distribution (Eq. (3.2)) can mathematically be expressed as a scalar product of two vectors  $\boldsymbol{\varphi}_i, \mathbf{f} \in \mathbb{R}^{19}$ . The moments of  $\mathbf{m}$  are chosen in a way that the 19 basis vectors  $\boldsymbol{\varphi}_i$  are an orthogonal basis of the moment space. Correspondingly, the moment vector can be written as:

$$\mathbf{m} = (\boldsymbol{\varphi}_0 \cdot \mathbf{f}, \boldsymbol{\varphi}_1 \cdot \mathbf{f}, \dots, \boldsymbol{\varphi}_{18} \cdot \mathbf{f})^T = \mathbf{M} \mathbf{f}. \quad (3.33)$$

$\mathbf{M}$  is a 19x19 matrix and holds the basis vectors  $\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_{18}$  as row entries (see Appendix A for a full length description of  $\mathbf{M}$  and the definition of  $\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_{18}$ ). So, the matrix  $\mathbf{M}$  assigns every distribution vector  $\mathbf{f}$  the according moments. Since the matrix  $\mathbf{M}$  has full rank, it is invertible and the matrix  $\mathbf{M}^{-1}$  can be used for the re-transformation from the moment space into velocity space.

The equilibrium state of the moments  $\mathbf{m}^{eq}$ , to which the relaxation is performed, is usually given by:

$$\mathbf{m}^{eq} = \mathbf{M}\mathbf{f}^{eq}. \quad (3.34)$$

with  $\mathbf{f}^{eq}$  according to Eq. (3.8). Since  $\mathbf{f}^{eq}$  is depending solely on density and velocity, all equilibrium moments  $m_i^{eq}$  are describable in terms of  $\rho$  and  $\mathbf{u}$ , see Eq. (A.1). In principle, other forms of the equilibrium moments are possible, as long as the collisional invariants are preserved [29]. Since the components of the momentum flux tensor are included in the moment vector  $\mathbf{m}$ , the macroscopic stress tensor  $\boldsymbol{\sigma}$  is easily computed from the moments of  $\mathbf{m}$  [37, 126] as shown in Eq. (A.3).

We focus now on the matrix  $\hat{\mathbf{S}} = \text{diag}(s_0, s_1, \dots, s_{18})$ , which contains the 19 relaxation times for the different moments. By a Chapman-Enskog expansion of the MRT-LBE (Eq. (3.31)), a relation between the different relaxation parameters  $s_i$  and the macroscopic transport coefficients can be established [27, 95]. The result of this analysis is that the relaxation parameters  $s_9, s_{11}, s_{13}, s_{14}, s_{15}$  depend on the kinematic viscosity by Eq. (3.15), what is obvious since they relax moments related to momentum fluxes. The parameter  $s_1$  is related to the bulk viscosity ( $\zeta$ ) of the fluid [29, 46, 73]. For the collisional invariants density and momentum, the parameters  $s_0, s_3, s_5, s_7$  are arbitrary since  $m_i = m_i^{eq}$ ,  $i \in \{0, 3, 5, 7\}$ . Commonly, they are chosen to 0 for simplicity. The remaining relaxation times are free parameters and are grouped, depending on the physical relation of the according moments:

$$\begin{aligned} s_0 &= s_3 = s_5 = s_7 = 0 \\ s_1 &= s_e \\ s_2 &= s_\epsilon \\ s_4 &= s_6 = s_8 = s_q \\ s_{10} &= s_{12} = s_\pi \\ s_9 &= s_{11} = s_{13} = s_{14} = s_{15} = s_\Omega = \Omega \\ s_{16} &= s_{17} = s_{18} = s_m. \end{aligned} \quad (3.35)$$

All relaxation parameters, except of  $s_\Omega = \Omega$  can be freely chosen in the range  $0 < s_i < 2$  for incompressible, isothermal flows. [72]. These relaxation times are normally adapted to improve stability and accuracy of the simulation. The optimal values can be determined through a linear analysis [72]. But since these optimal values are flow and geometry dependent, the free values  $s_i$  are commonly chosen to 1, which means that the according moments are relaxed directly to equilibrium. This approach is also known as regularized scheme [13, 74] and can be interpreted as a Ehrenfest step for the non-hydrodynamic quantities [13]. The SRT model can be interpreted

as a particular case of the MRT model presented here. By setting all relaxation times  $s_i = \Omega$ , the MRT model reduces to the SRT model [29]. Another special case of the MRT is derived by grouping the relaxation parameters for the even ( $s_1 = s_2 = s_9 = s_{10} = s_{11} = s_{12} = s_{13} = s_{14} = s_{15}$ ) and odd ( $s_4 = s_6 = s_8 = s_{16} = s_{17} = s_{18}$ ) moments of powers in  $\xi$ . The model received in this way is known as two relaxation time model [45]. Beside this intuitive derivation and description of the MRT model, a formal derivation of the MRT model from the Boltzmann equation by means of Hermite polynomials is given in literature by Shan and Chen [109].

The main advantage of the MRT model is the drastically increased stability compared to the SRT model. By relaxing the non-hydrodynamic moments directly to equilibrium, the according modes are damped quickly, what effects the higher stability [83, 95]. The disadvantage of this scheme is the higher computational effort. But, by carefully coding, the overhead can be limited to approximately 20 percent, compared to the SRT model [29].

### 3.4 Treatment of External Forces

External forces were, for simplicity, neglected in the derivation of the Lattice Boltzmann equation. The treatment of external forces in the Lattice Boltzmann Method will be considered afterwards. Since the treatment of forcing terms in the SRT and MRT model is a bit different, a separate introduction according to [47, 48] is given here.

#### 3.4.1 Single Relaxation Time Model

The presence of a body force is considered in the Lattice Boltzmann equation by an additional source term [47]:

$$f_i(\mathbf{x} + \xi_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \Omega \left[ f_i^{eq}(\mathbf{x}, t) - f_i(\mathbf{x}, t) \right] + \Delta t F_i. \quad (3.36)$$

Additionally, the first order moment for fluid momentum is redefined:

$$\rho \mathbf{u}^* = \sum_{i=0}^{q-1} \xi_i f_i + m \mathbf{f} \Delta t. \quad (3.37)$$

In Eq. (3.37), the term  $\mathbf{f}$  identifies the force per unit volume  $\mathbf{f} = \rho \mathbf{a}$ , where  $\mathbf{a}$  is the acceleration according to Newton's second law. Consequently, the equilibrium distribution (Eq. (3.8)) for the evaluation of Eq. (3.36) is performed with  $\mathbf{u}^*$ .

The introduced forcing term in  $F_i$  in Eq. (3.36) is expanded as a power series in  $\xi_i$  [47, 69]:

$$F_i = w_i \left[ A + \frac{\mathbf{b} \cdot \xi_i}{c_s^2} + \frac{\mathbf{C} : (\xi_i \xi_i - c_s^2 \mathbf{I})}{2c_s^4} \right]. \quad (3.38)$$

To get a valid model, the parameters  $A$ ,  $\mathbf{b}$ ,  $\mathbf{C}$  and  $m$  in the above equations need to be defined in a way that the hydrodynamic equations are correctly recovered. The tool for this derivation of

the parameters is again the Chapman-Enskog expansion. According to Guo et al. [47], the parameters must be chosen in the following way, to match the Navier-Stokes equations correctly:

$$\begin{aligned} A &= 0 & \mathbf{b} &= \frac{\Omega\nu}{c_s^2\Delta t}\mathbf{f} \\ C &= \frac{\Omega\nu}{c_s^2\Delta t}(\mathbf{u}^*\mathbf{f} + \mathbf{f}\mathbf{u}^*) & m &= \frac{1}{2}. \end{aligned} \quad (3.39)$$

With the parameters given above, Eqs. (3.39) and (3.37) result in the subsequent equations. Clearly, the influence of the discrete lattice effect on the forcing term can be seen.

$$F_i = w_i \frac{\Omega\nu}{c_s^2\Delta t} \left[ \frac{\xi_i - \mathbf{u}^*}{c_s^2} + \frac{\xi_i \cdot \mathbf{u}^*}{c_s^4} \xi_i \right] \cdot \mathbf{f} \quad (3.40)$$

$$\rho\mathbf{u}^* = \sum_{i=0}^{q-1} \xi_i f_i + \frac{1}{2}\mathbf{f}\Delta t \quad (3.41)$$

The expression for the force term (Eq. (3.40)) can also be deduced a priori in the derivation of the Lattice Boltzmann equation from the Boltzmann equation. This is done by a Hermite expansion of the forcing term in Eq. (2.11) [82, 111].

### 3.4.2 Multiple Relaxation Time Model

Guo et al. [48] adapt the above procedure for the SRT model to the MRT model. Since the discrete lattice effect depends on the relaxation parameter, this step must be considered in moment space for the MRT model. The generalized Lattice Boltzmann equation with forcing terms reads, in analogy to the SRT model, as:

$$f_i(\mathbf{x} + \xi_i\Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \sum_{j=0}^{q-1} (\mathbf{M}^{-1}\hat{\mathbf{S}}\mathbf{M})_{ij} [f_j^{eq} - f_j] + \Delta t \hat{F}_i. \quad (3.42)$$

The forcing term  $\hat{F}_i$  is the  $i$ -th component of the source term vector  $\hat{\mathbf{f}} \in \mathbb{R}^{19}$ :

$$\hat{\mathbf{f}} = \mathbf{M}^{-1} \left( \mathbf{I} - \frac{1}{2}\mathbf{S} \right) \mathbf{M}\bar{\mathbf{f}}. \quad (3.43)$$

where the discrete lattice effects act in the moment space here. The vector  $\bar{\mathbf{f}} \in \mathbb{R}^{19}$  contains the effect of the force on the distributions without discrete lattice effects and is derived in the same way as for the SRT model.  $\bar{\mathbf{f}}$  consists out of the following components:

$$\bar{F}_i = w_i \left[ \frac{\xi_i - \mathbf{u}^*}{c_s^2} + \frac{\xi_i \cdot \mathbf{u}^*}{c_s^4} \xi_i \right] \cdot \mathbf{f}. \quad (3.44)$$

An important point for the choice of the relaxation times belonging to fluid momentum is given by Eq. (3.43). Since the fluid momentum in the MRT scheme with external forces is defined according to Eq. (3.41), the relaxation parameters  $s_3, s_5, s_7$  must be chosen to 1 to end up in the mentioned definition.

With this forcing term, the MRT scheme recovers the Navier-Stokes equations correctly, as shown by [46, 48].

## 3.5 Boundary Conditions

Modeling of fluid flows requires appropriate conditions at the domain boundaries. In this section we will see how boundary conditions are realized in the Lattice Boltzmann Method. Generally, boundary conditions are specified by macroscopic values, which have to be transferred to values of the distribution function for usage in LBM. Our target in applying boundary conditions is to provide a valid set of distribution functions for all nodes in the fluid domain adjacent to boundaries. These distribution values must realize the macroscopic values specified at the boundary. We will restrict ourselves here to the boundary conditions implemented in SAM-Lattice and used in this thesis. A comprehensive overview of existing boundary conditions in the Lattice Boltzmann context is given by [46].

### 3.5.1 Equilibrium Schemes

The equilibrium distribution (Eq. (3.8)) gives a relation between the distribution functions and the hydrodynamic variables. So, it is evident to set the missing distribution functions by the equilibrium values, build from the macroscopic values at the boundary. Oftentimes, only one hydrodynamic value, pressure or velocity, is specified at a boundary. In such cases the remaining physical quantity is extrapolated normal to the boundary from the fluid. Setting of equilibrium values results in general in a first order accurate boundary condition [53].

There are some ideas for improving the accuracy of the scheme. This is done by adding a non-equilibrium part to the equilibrium at the boundary. Such models are then no longer named equilibrium schemes, in literature they are sometimes called extrapolation schemes instead. The non-equilibrium part can be approximated by the stress tensor, if this quantity is known at a boundary, as suggested from Latt [74]. But in most cases the stress tensor is unknown, what requires other techniques. A simple example is again the extrapolation of the non-equilibrium part from the fluid normal to wall. For further schemes, the reader is referred to [46].

### 3.5.2 Periodic Boundary Conditions

Periodic boundary conditions create an infinitely large domain, in the direction in which they are applied. Descriptively spoken, the periodic sides of the domain are linked together. In the Lattice Boltzmann context this is achieved by filling in the molecules, which are leaving the

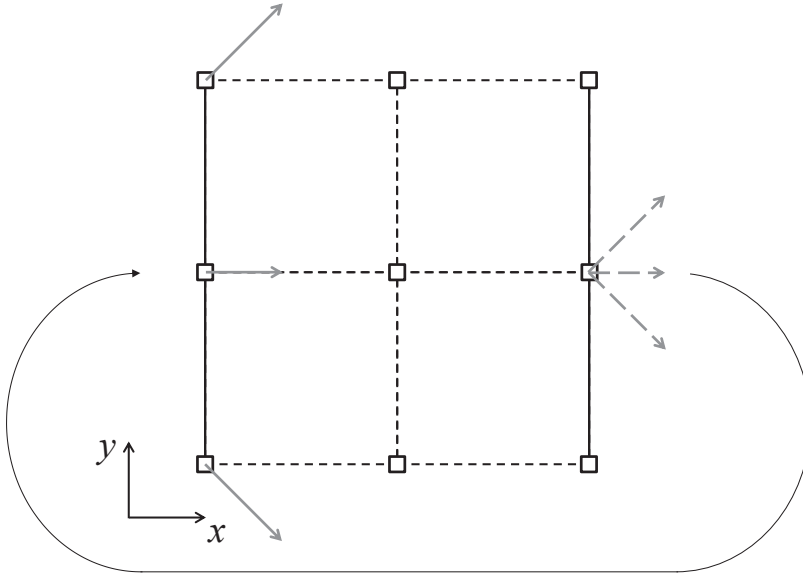


Figure 3.3: Periodicity Condition

domain at one side, in the opposite site of the domain. This is schematically shown in 2D in Fig. 3.3. The molecules leaving the domain, for instance in  $+x$  direction, are advected to the according neighbors at the domain beginning on the left hand side.

### 3.5.3 Bounce Back Schemes

An intuitive way to model solid walls is by molecular reflection. This is commonly known in the Lattice Boltzmann literature as *bounce back*. The physical meaning behind the bounce back rule is easily explained: Molecules colliding with a solid, impermeable wall are bounced back to the opposite direction of where they come from. Schematically, this is shown in 2D for one velocity direction in Fig. 3.4(a). The molecules flying in  $+x + y$  direction, which is direction number 16 in the  $D3Q19$  lattice, hit the solid wall and are bounced to direction 17 ( $-x - y$ ). Mathematically, the bounce back rule is expressed by:

$$f_i(\mathbf{x}_f, t + \Delta t) = f_i(\mathbf{x}_f, t) \quad \text{with} \quad \xi_i = -\xi_i. \quad (3.45)$$

for all directions  $\hat{i}$  with  $\xi_i \cdot \mathbf{n} < 0$ , i.e.,  $\xi_i$  pointing from the wall into the fluid. Since the introduced Lattice Boltzmann scheme is designed in a way that molecules or the according distribution values move from one node to a next neighbor node in one time step (i.e. convective scaling), the bounce back scheme has some geometrical constraints on the wall. The wall must be placed half a grid spacing away from the fluid nodes adjacent to the domain boundary, e.g.,  $\mathbf{x}_f$  in Fig. 3.4(a), to allow the molecules to hit the wall and fly back to the starting node in one



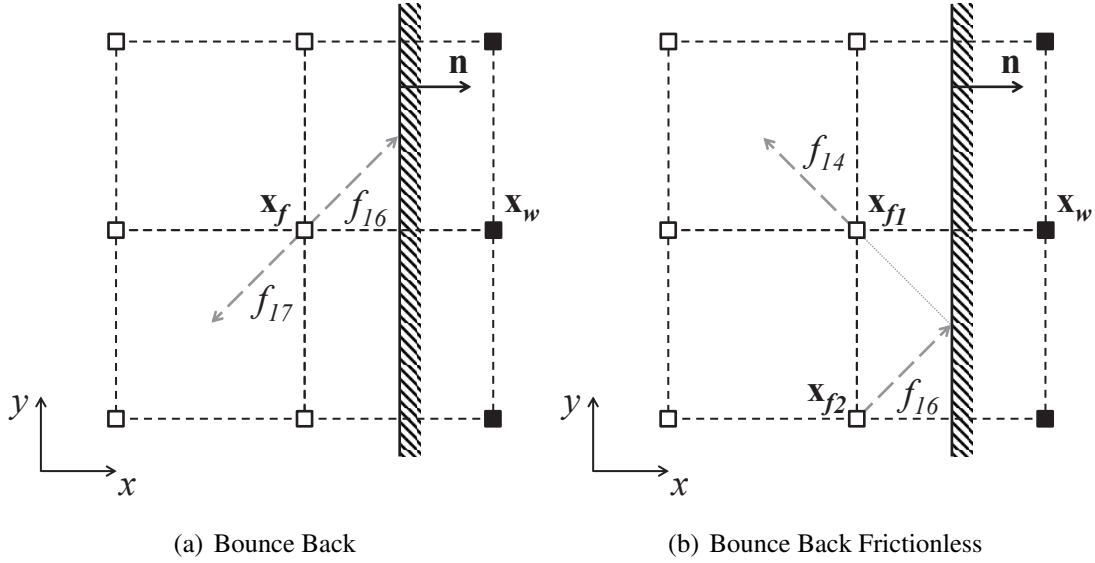


Figure 3.4: Bounce Back Schemes

time step. In other words, the wall must be located half way between the fluid node  $\mathbf{x}_f$  and the first node outside of the fluid domain, called wall node  $\mathbf{x}_w$ , of the equally spaced grid. Therefore, this boundary condition is often called half way bounce back. An implication of this constraint is that the wall must be flat.

The macroscopic behavior of the boundary can be determined from the fluxes through the wall. As shown by Chen et al. [18], mass and momentum fluxes vanish exactly by usage of the bounce back scheme and a impermeable, no-slip wall is achieved. To model moving boundaries with a wall velocity  $\mathbf{u}_w$ , Ladd [70] introduced an additional term in the bounce back scheme:

$$f_i(\mathbf{x}_f, t + \Delta t) = f_i(\mathbf{x}_f, t) - 2w_i\rho(\mathbf{x}_f, t) \frac{\boldsymbol{\xi}_i \cdot \mathbf{u}_w}{c_s^2}. \quad (3.46)$$

For flat walls, located half a spacing from the fluid nodes adjacent to the domain boundary, the bounce back rule is spatially second order accurate [53]. If the actual wall position is different from  $\Delta x/2$ , a discretization error of order  $\mathcal{O}(\Delta x)$  is introduced and thus the spatial accuracy reduces to first order. The same holds for curved boundaries, which degenerate to some stair step shape by application of the bounce back rule [50].

There is another kind of bounce back schemes, called specular reflection. In this scheme the molecules are reflected specular to the fluid, which means that the normal momentum of the molecules hitting the wall is reflected and the tangential momentum remains unchanged, see Fig. 3.4(b). This is mathematically expressed by:

$$f_i(\mathbf{x}_{f_1}, t + \Delta t) = f_i(\mathbf{x}_{f_2}, t) \quad \text{with} \quad \boldsymbol{\xi}_i = \boldsymbol{\xi}_i - 2(\boldsymbol{\xi}_i \cdot \mathbf{n}) \cdot \mathbf{n}. \quad (3.47)$$

Due to this treatment, the tangential momentum remains unchanged and a frictionless impermeable wall is modeled [46]. This frictionless bounce back has the same geometrical requirements to the wall as the standard half way bounce back scheme. Therefore, its usage is advisable only for flat walls, located half way between fluid and wall nodes since the scheme is second order accurate only in this case.

### 3.5.4 Bouzidi Scheme

The limitations of the bounce back schemes require advanced boundary conditions, especially for curved boundary shapes. A very elegant scheme was developed by Bouzidi et al. [11], which is an interpolation bounce back scheme. As we have seen in the previous subsection, for the standard bounce back rule the boundary position is forced half way between fluid and wall nodes. If the boundary position is different, there are direct consequences for the advection of molecules.

For the following considerations, the boundary position is distinguished into two cases: The boundary can be located half a spacing or more away from a fluid node  $\mathbf{x}_{f_1}$  (see Fig. 3.5(a)) or the boundary is placed closer than  $\Delta x/2$  to a fluid node  $\mathbf{x}_{f_1}$  in Fig. 3.5(b). The distance between the boundary and the fluid node is measured by the parameter  $q$ , see Eq. (3.48). In the first case we have  $q \geq 0.5$  and in the second one  $q < 0.5$ .

$$q = \frac{x_f - x_{\text{boundary}}}{\Delta x} \quad (0 < q \leq 1) \quad (3.48)$$

In the case  $q \geq 0.5$ , molecules or distribution functions, respectively, leaving at the node  $\mathbf{x}_{f_1}$  at time  $t$ , for instance in  $x$  direction, will hit the boundary and be reflected (bounced back) in  $-x$  direction. Since  $q \geq 0.5$ , the boundary is too far away for the molecules to reach  $\mathbf{x}_{f_1}$  in one time step  $\Delta t$ . At the end of the time step ( $t + \Delta t$ ) the molecules reside at  $\mathbf{x}_s$ , which is at a distance  $s = \Delta x(2q - 1)$  away from the fluid node  $\mathbf{x}_{f_1}$ , see Fig. 3.5(a). To get a valid distribution function at the fluid node  $\mathbf{x}_{f_1}$  at  $t + \Delta t$ , the distribution function is linearly interpolated from the known values at the node  $\mathbf{x}_{f_2}$  and at the end position of the bounced back molecules  $\mathbf{x}_s$ . The same approach is tracked for  $q < 0.5$ , see Fig. 3.5(b). Molecules leaving  $\mathbf{x}_{f_1}$  at time  $t$  will be reflected in  $-x$  direction at the wall and reside at  $\mathbf{x}_s$  at the end of the time step ( $t + \Delta t$ ). Now this position is located between  $\mathbf{x}_{f_1}$  and  $\mathbf{x}_{f_2}$ , since  $q$  is less than 0.5. The calculation of the distribution function at  $\mathbf{x}_{f_1}$  in  $-x$  direction at time  $t + \Delta t$  would result in an extrapolation if it is performed in the same fashion as for  $q \geq 0.5$ . Numerically seen, an extrapolation can generate instability. To avoid this, the basic idea is used the other way around. Molecules starting at  $\mathbf{x}_s$  at time  $t$ , are reflected at the wall and will arrive exactly at  $\mathbf{x}_{f_1}$  at the end of the time step ( $t + \Delta t$ ). So, the distribution function is interpolated linearly to  $\mathbf{x}_s$  from  $\mathbf{x}_{f_1}$  and  $\mathbf{x}_{f_2}$  at time  $t$  and then the advection and bounce back is performed to get a valid distribution function at the fluid node  $\mathbf{x}_{f_1}$  at  $t + \Delta t$ .

Mathematically, the Bouzidi scheme can be expressed by [11]: (notice  $\xi_i = -\xi_i$ )

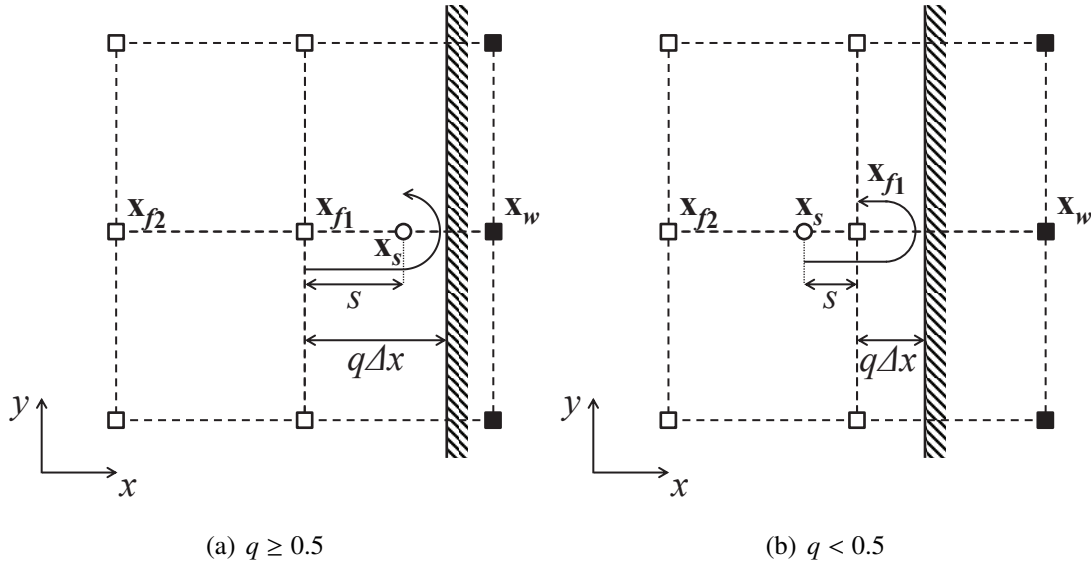


Figure 3.5: Bouzidi Scheme

$$f_i(\mathbf{x}_{f_1}, t + \Delta t) = \begin{cases} \frac{1}{2q_i} f_i(\mathbf{x}_{f_1}, t) + \frac{2q_i - 1}{2q_i} f_i(\mathbf{x}_{f_2}, t + \Delta t) - \frac{1}{q_i} w_i \rho(\mathbf{x}_{f_1}, t) \frac{\xi_i \cdot \mathbf{u}_w}{c_s^2} & q_i \geq 0.5 \\ 2q_i f_i(\mathbf{x}_{f_1}, t) + (1 - 2q_i) f_i(\mathbf{x}_{f_2}, t) - 2w_i \rho(\mathbf{x}_{f_1}, t) \frac{\xi_i \cdot \mathbf{u}_w}{c_s^2} & q_i < 0.5 \end{cases} \quad (3.49)$$

The last term on the right hand side of Eq. (3.49) incorporates a potential wall velocity  $\mathbf{u}_w$ . For  $q = 0.5$ , the Bouzidi scheme results exactly in the bounce back rule. By applying this scheme for all boundary adjacent fluid nodes and all directions pointing to the boundary (see Fig. 3.6), arbitrary shaped velocity boundaries can be modeled with second order spatial accuracy [11].

A very important result of this gas kinetic treatment of boundary conditions is the fact that no body conformal grids are needed in LBM to model arbitrary geometrical shapes. The fluid domain is meshed with an equally spaced rectangular grid, and the shape of the domain boundaries is implicitly included in the values of  $q_i$ .

### 3.5.5 Yu Scheme

Another interpolation based bounce scheme was proposed by Yu et al. [137]. In the Yu scheme a single interpolation rule is used, independent of the boundary distance  $q_i$ . The basic idea of Yu is summarized in the following with help of Fig. 3.6: The boundary is assumed to be permeable so that the molecules can pass through. If we look at the molecules flying from  $\mathbf{x}_{f_1}$  in 0-direction (+x) to the boundary, they will cross the boundary after a distance of  $q_0 \Delta x$  and arrive at the wall node  $\mathbf{x}_w$  at the end of a time step  $t + \Delta t$ . The distribution in 0-direction on the boundary, i.e.,  $\mathbf{x}_b = \mathbf{x}_{f_1} + \xi_0 q_0 \Delta t$ , at time  $t + \Delta t$  is then linearly interpolated from the values at

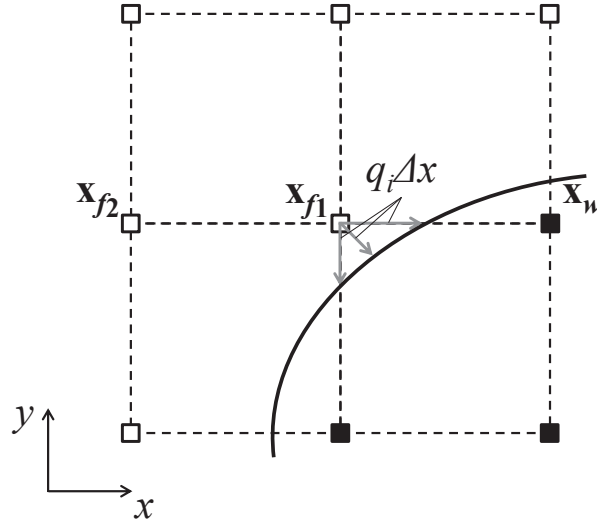


Figure 3.6: Treatment for Curved Boundaries

$\mathbf{x}_{f_1}$  and  $\mathbf{x}_w$ . Generally written, this is expressed by:

$$f_i(\mathbf{x}_b, t + \Delta t) = f(\mathbf{x}_{f_1}, t + \Delta t) + q_i \left( f_i(\mathbf{x}_w, t + \Delta t) - f_i(\mathbf{x}_{f_1}, t + \Delta t) \right). \quad (3.50)$$

The interpolated distribution at the boundary is subsequently bounced back. To assign a wall velocity  $\mathbf{u}_w$  the already known term of Eq. (3.45) is used:

$$f_i(\mathbf{x}_b, t + \Delta t) = f_i(\mathbf{x}_b, t + \Delta t) - 2w_i \rho(\mathbf{x}_{f_1}, t) \frac{\xi_i \cdot \mathbf{u}_w}{c_s^2}. \quad (3.51)$$

The missing distribution function at the fluid node  $\mathbf{x}_{f_1}$  at time  $t + \Delta t$  is obtained by a linear interpolation of the distribution functions in  $\hat{i}$ -direction at the boundary (Eq. (3.51)) and at the neighbor node  $\mathbf{x}_{f_2}$ :

$$f_i(\mathbf{x}_{f_1}, t + \Delta t) = f_i(\mathbf{x}_b, t + \Delta t) + \frac{q_i}{1 + q_i} \left( f_i(\mathbf{x}_{f_2}, t + \Delta t) - f_i(\mathbf{x}_b, t + \Delta t) \right). \quad (3.52)$$

As we can see from the preceding equations, the Yu scheme does not reduce to the bounce back rule in the case of  $q_i = 0.5$ . Nevertheless, this scheme is also second order accurate, as shown in [137].

Although the physical ideas of the Yu and the Bouzidi schemes are different, they are, from a mathematical point of view, members of the same class of linear interpolation schemes. They differ only in their interpolation coefficients, as shown by Ginzburg et al. [45], who introduce further schemes of this class.

### 3.5.6 Mei Scheme

Besides the interpolation bounce back schemes discussed in the last two subsections, there is another type of boundary schemes for arbitrarily shaped velocity boundaries. These schemes are sometimes named fictitious equilibrium schemes. The method proposed by Mei et al. [85] is implemented in SAM-Lattice and a representative of this group. The Mei scheme is based on the work of [35, 84].

In the fictitious equilibrium schemes, the missing distribution functions are linearly interpolated from the standard bounce back value and a fictitious equilibrium [85]:

$$f_i(\mathbf{x}_f, t + \Delta t) = (1 - \chi)f_i(\mathbf{x}_f, t) + \chi f_i^{eq,*}(\mathbf{x}_w, t) - 2w_i\rho(\mathbf{x}_f, t)\frac{\xi_i \cdot \mathbf{u}_w}{c_s^2}. \quad (3.53)$$

Again  $\hat{i}$  is the counter direction of  $i$  in the lattice ( $\xi_{\hat{i}} = -\xi_i$ ) and the last term on the right hand side of Eq. (3.53) incorporates a potential boundary velocity  $\mathbf{u}_w$ . The fictitious equilibrium  $f_i^{eq,*}$  is defined as:

$$f_i^{eq,*}(\mathbf{x}_w, t) = w_i\rho(\mathbf{x}_f, t) \left\{ 1 + \frac{(\xi_i \cdot \mathbf{u}_{bf})}{c_s^2} + \frac{(\xi_i \cdot \mathbf{u}(\mathbf{x}_f, t))^2}{2c_s^4} - \frac{\mathbf{u}^2(\mathbf{x}_f, t)}{2c_s^2} \right\}. \quad (3.54)$$

The interpolation parameter  $\chi$  and the velocity  $\mathbf{u}_{bf}$  are chosen as a function of the distance to the boundary  $q_i$  for the lattice direction  $i$ , see Fig. 3.6. Like the interpolation schemes, the fictitious equilibrium schemes do neither need body conformal grids to model curved boundaries.

$$\mathbf{u}_{bf} = \begin{cases} \left(1 - \frac{3}{2q_i}\right)\mathbf{u}(\mathbf{x}_f, t) + \frac{3}{2q_i}\mathbf{u}_w & q_i \geq 0.5 \\ \mathbf{u}(\mathbf{x}_f + \xi_i\Delta t, t) & q_i < 0.5 \end{cases} \quad (3.55)$$

$$\chi = \begin{cases} \frac{2q_i-1}{\Omega+0.5} & q_i \geq 0.5 \\ \frac{2q_i-1}{\Omega-2} & q_i < 0.5 \end{cases} \quad (3.56)$$

Equation (3.55) shows that the velocity  $\mathbf{u}_{bf}$  is interpolated between the velocity at the fluid node  $\mathbf{x}_f$  and the boundary velocity  $\mathbf{u}_w$  for  $q_i \geq 0.5$ . In the case  $q_i < 0.5$  the velocity at the next neighbor node of  $\mathbf{x}_f$  in  $\hat{i}$  direction is used. The interpolation parameter  $\chi$  is defined piecewise for  $q_i$ , too. It can be determined by a Chapman-Enskog expansion [84]. The resulting  $\chi$  is a function of  $q_i$  and the relaxation parameter  $\Omega$ , given in Eq. (3.56).

As shown by [84, 85], this boundary treatment is second order accurate for arbitrarily shaped boundaries.

### 3.5.7 Shear Boundaries

All boundary conditions discussed so far are of Dirichlet type, which means that values of macroscopic quantities are specified on the boundary. In some cases the derivative of a quantity must be set on a boundary. This is commonly known as Neumann boundary condition. An

example of a Neumann condition is the specification of the wall shear stress at a boundary, since the shear stress is related to the derivative of the velocity by:

$$\boldsymbol{\tau}_w = \mu \frac{\partial \mathbf{u}_t}{\partial \mathbf{n}}. \quad (3.57)$$

The literature on shear stress boundary conditions in LBM is rather limited. Succi [116] suggests to use a linear combination of bounce back (Eq. (3.45)) and specular reflection (Eq. (3.47)), what is often called slip-reflection model. But in this approach there are no general mathematical correlations between the coefficients of the linear combination and the wall shear stress. This drawback limits the utilization strongly.

Other known schemes for Neumann conditions are hydrodynamic schemes based on the inter- or extrapolation of the Dirichlet value on the boundary, like it is done in classical finite difference methods, see, e.g., [34]. For the shear stress, this means that the velocity on the boundary (Dirichlet value) is inter- or extrapolated from the bulk in a way to match the desired shear stress and velocity derivative, respectively. With the velocity determined in this way, one of the already introduced Dirichlet boundary conditions, like the Bouzidi, Yu or Mei scheme can be applied to finally impose the boundary condition. This procedure was proposed by Izquierdo and Fuego [59] and will be presented in the following. In the remaining part of the thesis, this boundary condition will be called shear bounce back scheme.

As said before, the first step in realizing the shear bounce back scheme is the determination of the velocity at the wall to meet the desired velocity gradient. Since the wall shear stress  $\boldsymbol{\tau}_w$  is related to the derivative of the wall tangential velocity  $\mathbf{u}_t$  in wall normal direction  $\mathbf{n}$  (see Eq. (3.57)) it is convenient to use here a rotated coordinate system in which the wall normal is a basis vector, cf. Fig. 3.7. To determine the velocity at the wall, a Taylor expansion of the tangential velocity at the points  $\mathbf{x}_{I_1}$  and  $\mathbf{x}_{I_2}$  is performed:

$$\mathbf{u}_t(\mathbf{x}_{I_1}) = \mathbf{u}_t(\mathbf{x}_b + h\mathbf{n}) = \mathbf{u}_t(\mathbf{x}_b) + h \frac{\partial \mathbf{u}_t}{\partial \mathbf{n}}(\mathbf{x}_b) + \frac{h^2}{2} \frac{\partial^2 \mathbf{u}_t}{\partial \mathbf{n}^2}(\mathbf{x}_b) + \mathcal{O}(h^3) \quad (3.58)$$

$$\mathbf{u}_t(\mathbf{x}_{I_2}) = \mathbf{u}_t(\mathbf{x}_b + 2h\mathbf{n}) = \mathbf{u}_t(\mathbf{x}_b) + 2h \frac{\partial \mathbf{u}_t}{\partial \mathbf{n}}(\mathbf{x}_b) + \frac{(2h)^2}{2} \frac{\partial^2 \mathbf{u}_t}{\partial \mathbf{n}^2}(\mathbf{x}_b) + \mathcal{O}(h^3). \quad (3.59)$$

The point  $\mathbf{x}_b$  on the wall is the intersection point of the boundary and the lattice link in the direction which is missing. The Taylor series are expanded up to second order, since LBM is second order accurate. The distance in normal direction between  $\mathbf{x}_{I_1}$ ,  $\mathbf{x}_{I_2}$ , and the wall  $\mathbf{x}_b$  is  $h$  and  $2h$ , respectively. The choice of  $h$  is somewhat arbitrary, but it is common to choose  $h = \Delta x$ . From these two expansions (Eqs. (3.58), (3.59)) an equation for  $\mathbf{u}_t(\mathbf{x}_b)$  can be derived, when the terms of  $\mathcal{O}(h^3)$  are neglected:

$$\mathbf{u}_t(\mathbf{x}_b) = \frac{4\mathbf{u}_t(\mathbf{x}_{I_1}) - \mathbf{u}_t(\mathbf{x}_{I_2}) - 2\Delta x \frac{\partial \mathbf{u}_t}{\partial \mathbf{n}}(\mathbf{x}_b)}{3}. \quad (3.60)$$

The same result is obtained if one fits a parabolic polynomial for  $\mathbf{u}_t(\mathbf{n})$  through the values at the points  $\mathbf{x}_{I_1}$  and  $\mathbf{x}_{I_2}$  and uses the derivative  $\frac{\partial \mathbf{u}_t}{\partial \mathbf{n}}$  at the wall as third constraint. Hence, this procedure will be called quadratic interpolation in the following. The remaining step for solving Eq.

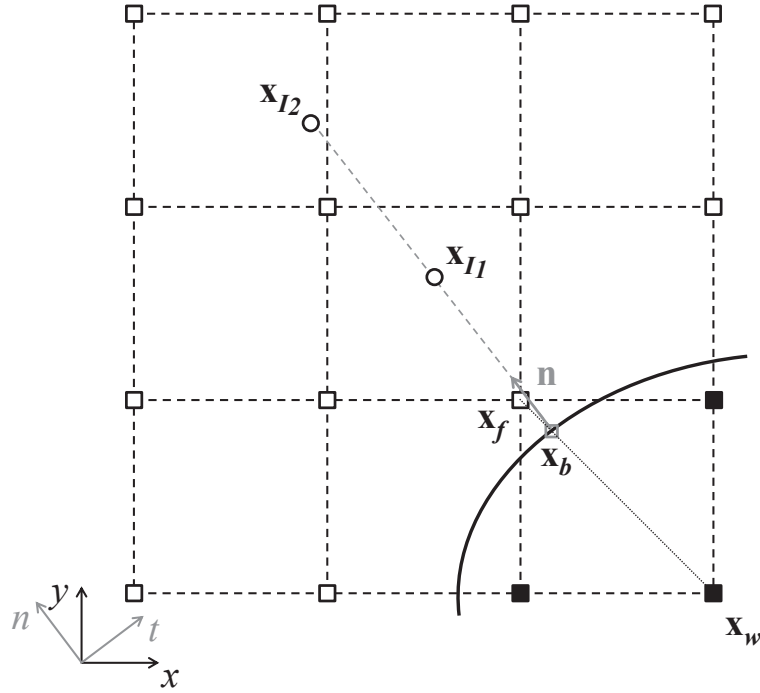


Figure 3.7: Interpolation for Shear Bounce Back Scheme

(3.60) is the calculation of  $\mathbf{u}_t(\mathbf{x}_{I_1})$  and  $\mathbf{u}_t(\mathbf{x}_{I_2})$ . To achieve a consistent scheme, the interpolation of these values should be second order accurate as well. Tölke and Krafczyk [125] and Geier et al. [42] proposed compact second order accurate interpolation methods. The advantage of these methods is that they work locally in one cell and are more accurate than bi-linear methods since they incorporate the local gradients at nodes which are available from the non-equilibrium part of the distribution functions. In SAM-Lattice a 3 dimensional extension of the compact interpolation method of [125] is used for this task.

The last steps for the realization of the shear bounce back scheme are to compose the velocity and perform one of the Dirichlet boundary conditions of this section. The velocity at the wall consists out of the tangential velocity (according to Eq. (3.60)), which is needed to impose the desired shear, and the velocity in normal direction, which is set to zero to avoid a mass flux, i.e., to ensure the impermeability of the boundary. As Dirichlet boundary condition the Yu scheme is used in the shear bounce back scheme in SAM-Lattice, because this scheme showed the best numerical stability of the second order accurate methods in combination with the interpolation procedure.

In contrast to the quadratic interpolation derived above, Izquierdo and Fueyo [59] use a linear interpolation for  $\mathbf{u}_t(\mathbf{x}_b)$ , which results from Eq. (3.58) for expansion up to first order:

$$\mathbf{u}_t(\mathbf{x}_b) = \mathbf{u}_t(\mathbf{x}_{I_1}) - \Delta x \frac{\partial \mathbf{u}_t}{\partial \mathbf{n}}(\mathbf{x}_b). \quad (3.61)$$

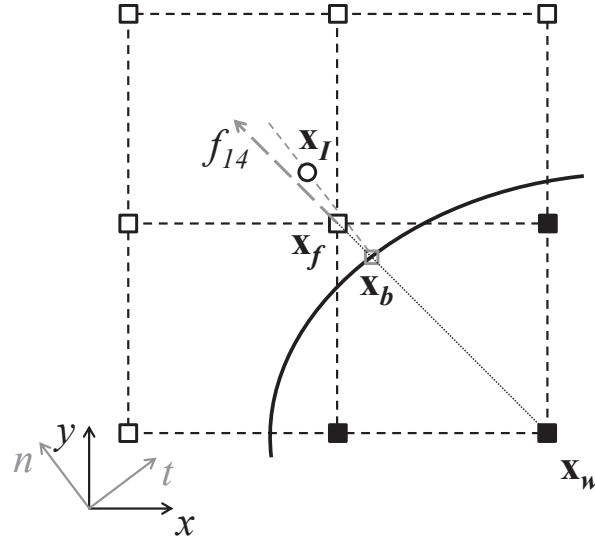


Figure 3.8: Interpolation Scheme for Shear Anti Bounce Back Scheme

An advantage of the linear scheme is the reduced computational effort, especially in parallel, distributed calculations. Disadvantageous is the reduced accuracy compared to Eq. (3.60).

Another way to impose shear boundary conditions known in literature is proposed by Chen et al. [18]. But the scheme of [18] uses a volumetric approach of the Lattice Boltzmann Method. In the following, the idea of [18] is adapted to create a shear boundary condition for the node based finite difference Lattice Boltzmann scheme presented in this thesis. This boundary condition is called shear anti bounce back scheme.

The first step in realizing the shear anti bounce back scheme is again the interpolation of the fluid velocity. But, in contrast to the shear bounce back, here the velocity is interpolated at a normal distance of half a spacing from the wall ( $\mathbf{x}_I$ ), see Fig. 3.8. For the velocity interpolation at off lattice positions again the 3 dimensional extension of [125] is used, as it is done for the shear bounce back scheme. The interpolated velocity  $\mathbf{u}_{int}$  at  $\mathbf{x}_I$  is decomposed into a wall normal  $\mathbf{u}_{int}^n$  and a wall tangential part  $\mathbf{u}_{int}^t$ . Next, the density at the wall  $\rho_w$  needs to be approximated. This is done by an extrapolation from the fluid node in the missing lattice direction  $\hat{i}$ , which needs to be constructed by the boundary condition (in Fig. 3.8:  $\mathbf{x}_f$ ).

The distribution in the missing lattice direction  $\hat{i}$  is a composition of different parts:

$$f_{\hat{i}}(\mathbf{x}_w, t) = f_{\hat{i}}(\mathbf{x}_f, t + \Delta t) = f_{\hat{i}}^{eq} + f_{\hat{i}}^{neq} + f_{\hat{i}}^{shear}. \quad (3.62)$$

$f_{\hat{i}}^{eq}$  is the equilibrium distribution (Eq. (3.8)) in the missing lattice direction, build with the tangential part of the interpolated velocity  $\mathbf{u}_{int}^t$  and the extrapolated density  $\rho_w$ . This assumes that there is no change of these hydrodynamic quantities in the vicinity of the wall. The non-



equilibrium part of the distribution function is achieved by the anti-bounce back rule [45]:

$$f_i^{neq} = -f_i^{neq} = f_i^{eq} - f_i(\mathbf{x}_f, t). \quad (3.63)$$

The non-equilibrium part of the distribution is set to the negative value of the incoming non-equilibrium distribution at the wall, therefore, it is called anti bounce back.  $f_i^{eq}$  is here the equilibrium distribution in the incoming direction, again build with  $\mathbf{u}_{int}^t$  and  $\rho_w$ . Relation (3.63) can only be exact if the boundary is half way between fluid and wall node, i.e.,  $q_i = 0.5$ . The remaining part  $f_i^{shear}$ , used to consider a tangential momentum difference, is neglected for the following analysis, i.e.,  $f_i^{shear} = 0$ , and will be defined later.

To evaluate the forces  $\mathbf{f}$  on a boundary in LBM, the momentum exchange methods can be used [44, 70, 86]. The momentum exchange method is based on Newton's second law and calculates the force by the change of the momentum per time step. If the momentum exchange method is applied for a surface element of the boundary, which is represented by a wall node, the resulting force on the surface element can be analyzed [18]:

$$\mathbf{f} = \frac{\Delta V}{\Delta t} \sum_{i \in C} [f_i(\mathbf{x}_w, t) + f_i(\mathbf{x}_w, t + \Delta t)] \xi_i. \quad (3.64)$$

This treatment is in principle only valid for  $q_i = 0.5$ , i.e., flat boundaries. For curved boundaries, interpolation techniques for the momentum exchange methods have to be considered [44]. If  $C$  is the set of all lattice links intersecting the boundary and  $\Delta V$  represents the volume of the cell surrounding the node, then:  $\Delta V = |\xi_i \cdot \mathbf{n}| A \Delta t$ . Inserting the Eqs. (3.62) and (3.63) in Eq. (3.64) yields to:

$$\begin{aligned} \mathbf{f} &= |\xi_i \cdot \mathbf{n}| A \sum_{i \in C} [f_i^{eq} + f_i^{eq} - f_i(\mathbf{x}_f, t) + f_i(\mathbf{x}_w, t + \Delta t)] \xi_i \\ \boldsymbol{\sigma} &= \frac{\mathbf{f}}{A} = |\xi_i \cdot \mathbf{n}| \sum_{i \in C} [f_i^{eq} + f_i^{eq} - f_i(\mathbf{x}_f, t) + f_i(\mathbf{x}_f, t)] \xi_i \\ \boldsymbol{\sigma} &= |\xi_i \cdot \mathbf{n}| \sum_{i \in C} (f_i^{eq} + f_i^{eq}) \xi_i \\ \boldsymbol{\sigma} &= \mathbf{n} \sum_{i \in C'} f_i^{eq} \xi_i \xi_i. \end{aligned} \quad (3.65)$$

The set  $C'$  in Eq. (3.65) consist of all  $i \in C$  and their counter directions  $\hat{i}$ . Equation (3.65) is the equilibrium part of the momentum flux tensor, cf. Eq. (3.20), and can be expressed by hydrodynamic quantities [50]:

$$\Pi_{\alpha\beta}^{eq} = \sum_i f_i^{eq} \xi_{i,\alpha} \xi_{i,\beta} = \rho u_\alpha u_\beta + p \delta_{\alpha\beta}. \quad (3.66)$$

Since the velocity  $\mathbf{u}_{int}^t$ , which was used to build the equilibrium, is tangential to the boundary, Eq. (3.66) reduces to:

$$\boldsymbol{\sigma} = \mathbf{n} p. \quad (3.67)$$

Thus, the resulting force on a surface element acts only in normal direction of the surface element and is created by the fluid pressure. In general mass conservation is not guaranteed by the above scheme. To balance the incoming and outgoing mass fluxes of the wall, Chen et al. [18] introduced a mass flux correction, which is transferred to the node based approach here. For every wall node, the difference between incoming and outgoing distributions (according to Eq. (3.62)) is averaged per energy level:

$$\delta f_j = \left( \sum_{\substack{i \in C, \\ |\xi_i| = \sqrt{j} \bar{\xi}}} 1 \right)^{-1} \left( \sum_{\substack{i \in C, \\ |\xi_i| = \sqrt{j} \bar{\xi}}} f_i^{in} - \sum_{\substack{i \in C, \\ |\xi_i| = \sqrt{j} \bar{\xi}}} f_i^{out} \right), \quad j \in \{1, 2\}. \quad (3.68)$$

The averaged mass correction per energy level is added to Eq. (3.62), where the parameter  $j$  is chosen according to the energy level ( $j = 1 \forall |\xi_i| = \bar{\xi}$  and  $j = 2 \forall |\xi_i| = \sqrt{2} \bar{\xi}$ ):

$$f_i(\mathbf{x}_f, t + \Delta t) = f_i^{eq} + f_i^{neq} + f_i^{shear} + \delta f_j. \quad (3.69)$$

The mass correction according to Eq. (3.68) ensures mass conservation, but introduces a change of the force and pressure on the surface element, respectively. From a momentum exchange performed with Eq. (3.69) the modified stress on the boundary element results in:

$$\boldsymbol{\sigma} = \mathbf{n}(p + \delta p). \quad (3.70)$$

Nevertheless, the modified stress or pressure, respectively, is still acting normal to the surface element. According to [18] the pressure difference  $\delta p$  is caused by a non-zero normal velocity, which arises from the mass correction.

The above derivation shows that for  $f_i^{shear} = 0$ , a frictionless surface is created. By  $f_i^{shear}$  the surface friction can be controlled. Chen et al. [18] proposed the following term to create a specific wall shear stress  $\tau_w$ , which is acting in the opposite direction of the interpolated tangential fluid velocity  $\mathbf{u}_{int}^t$ :

$$f_i^{shear} = \frac{\tau_w}{\rho c_s^2 |\mathbf{u}_{int}^t|} \mathbf{n} \cdot \boldsymbol{\xi}_i (f_i^{eq} - f_i^{eq}). \quad (3.71)$$

One can show that the desired friction is achieved by considering Eq. (3.71) in the momentum exchange applied to a surface element [18].

The shear anti bounce back scheme is expected to be only first order accurate in space for arbitrary flows, since this behavior was reported for the volumetric version of the scheme [77]. In contrast, the shear bounce back scheme should produce second order accurate solutions as long as all ingredients of the scheme are second order accurate. But, in practice the shear anti bounce back reveals better numerical stability, especially for high Reynolds number flows. The numerical accuracy of the different schemes for shear boundary schemes will be investigated numerically in Section 6.

## 3.6 Initial Conditions

From a mathematical point of view the Lattice Boltzmann equation as well as the Navier-Stokes equations are initial-boundary value problems. After the boundary treatment was discussed in the last section, now possibilities to specify initial conditions for the distribution values are presented.

### 3.6.1 Analytical Schemes

The easiest way to set initial conditions for  $f_i$  is using the equilibrium distribution values (Eq. (3.8)) build with initial values for density  $\rho_0$  and velocity  $\mathbf{u}_0$ .

To specify a non-equilibrium part of the distributions, the derivatives of the velocity field are required. With this information the non-equilibrium part can be initialized by [46]:

$$f_i^{neq} = -\Omega \Delta t w_i \rho_0 \left[ \frac{\xi_i \xi_i}{c_s^2} : \nabla \mathbf{u} - \nabla \cdot \mathbf{u} \right]. \quad (3.72)$$

This procedure for  $f_i^{neq}$  is not mandatory for steady or quasi-steady flows, since these flows are independent of the initial conditions; see [46] for details.

If the derivatives are not known analytically, they can be approximated by numerical stencils based on the initial values of  $\mathbf{u}_0$ , but this is no longer an analytical approach.

### 3.6.2 Iterative Schemes

The pressure field  $p_0$ , which is directly related to  $\rho_0$ , and the velocity field  $\mathbf{u}_0$  must be compatible in a way that the Navier-Stokes equations are fulfilled. If  $p_0$  or  $\rho_0$ , respectively, are not known analytically, they can be obtained for incompressible flows by solving a Poisson equation [46]. [15, 87] proposed schemes to solve the resulting Poisson equations in a LB like scheme. The scheme of Mei et al. [87] is implemented in SAM-Lattice, but its usage is strongly limited due to stability problems of the scheme itself. For this reason, the reader is referred to [87] for more details.

## 3.7 Local Grid Refinement

The LBM techniques discussed so far work only on regular, i.e., rectangular and equidistant, lattices. From numerical viewpoints it is desirable to use irregular grids. For instance, one might concentrate the grid in some regions, e.g., where high gradients in the flow occur, or reduce the grid density in other regions to save computational resources.

Different kinds of Lattice Boltzmann techniques for irregular grids have been developed since the origins of the method. As we have seen at the beginning of this section, LBM is a

special finite difference technique for solving the Boltzmann equation. From this point of view, other discretization methods of finite difference, finite volume or finite element type can be used. These methods work in general on irregular grids, but will also introduce higher computational and algorithmic effort. Since these types of methods are not used in SAM-Lattice, the reader is referred to [46] for more details. Another possibility to use irregular grids are the so called interpolation methods. Here, the computational mesh and the underlying lattice, based on the discrete velocity set, are decoupled. This means that the molecules are not flying from one node to their next neighbors in one timestep. Instead, the molecules are flying in the direction of the underlying lattice and end up at positions, which are not necessarily nodes of the computational mesh. From the values of the distribution functions at these end positions, the values at the computational nodes are interpolated. The interpolation methods are neither implemented in SAM-Lattice and will not be further discussed here, see [46] for an overview.

The use of irregular grids is provided by a grid refinement technique in SAM-Lattice. Local grid refinement was first proposed by Filippova and Hänel [35]. Details about the scheme implemented in SAM-Lattice will be given in the following subsections.

### 3.7.1 General Concept

The computational effectiveness of LBM strongly relies on the use of regular grids. To establish a scheme which enables the use of different grid resolutions while maintaining the efficiency, Filippova and Hänel [35] proposed to embed patches or blocks of a refined regular grid into a coarser regular grid. This basic idea has been advanced by different researchers [25, 31, 136]. The method used in SAM-Lattice is based on an octree grid, as proposed by [25, 37, 126]. Octree means that a regular coarse cell, called parent cell, is divided into eight regular fine cells, called child cells, see Fig. 3.9. According to this definition, the ratio between two subsequent grid levels is 2. The refinement of cells can be recursively continued until the desired refinement level or the corresponding mesh size, respectively, is achieved. A cell with the largest spacing  $\Delta x_0$  is named level 0 cell in the following. Hence a  $n^{\text{th}}$ -level cell has a spacing of  $\Delta x_n = \Delta x_0/2^n$ . To allow an efficient implementation of this approach, we restrict ourselves to smoothed octrees, what means that the level of neighboring cells can only differ by one, see ,e.g., [25]. Furthermore, a minimum width of three fine cells per grid level is required. This requirement is to prevent an overlapping of the mesh areas of the multiple levels, which need special treatment. The octree structure enables a hierarchical refinement of the computational domain and an efficient computational handling through advanced data structures.

A simple grid layout including local refinement is shown in two dimensions in Fig. 3.10. From an algorithmic point of view, the standard Lattice Boltzmann scheme can be applied on every grid level. Nevertheless, some specialties must be met.

Firstly, some distribution values are missing at the boundary of a grid level to another grid level, which is called interface. This boundary value problem is a bit different to the one at the domain boundaries. In contrast to the domain boundaries, the information of the neighboring level is available at the interfaces. As the hydrodynamic moments, i.e., density  $\rho$ , velocity  $\mathbf{u}$ ,

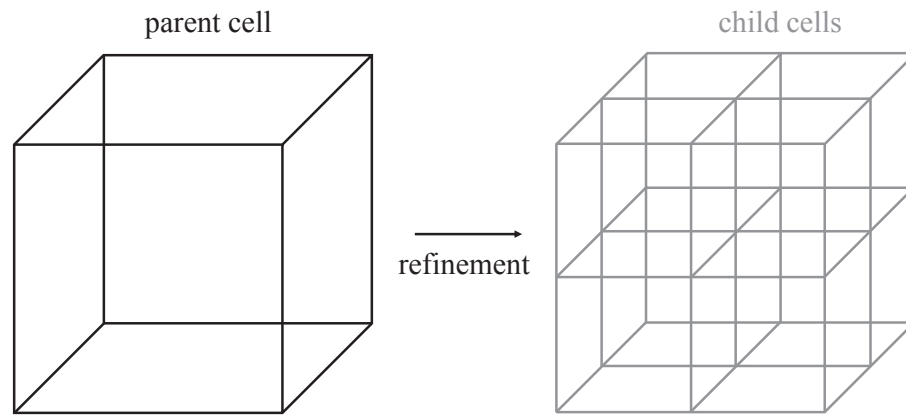


Figure 3.9: Octree Refinement

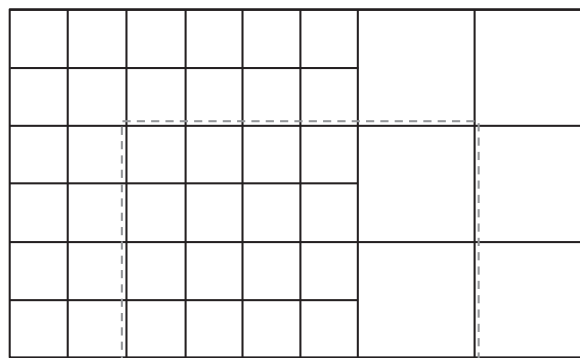


Figure 3.10: Simple Grid Layout (2D)

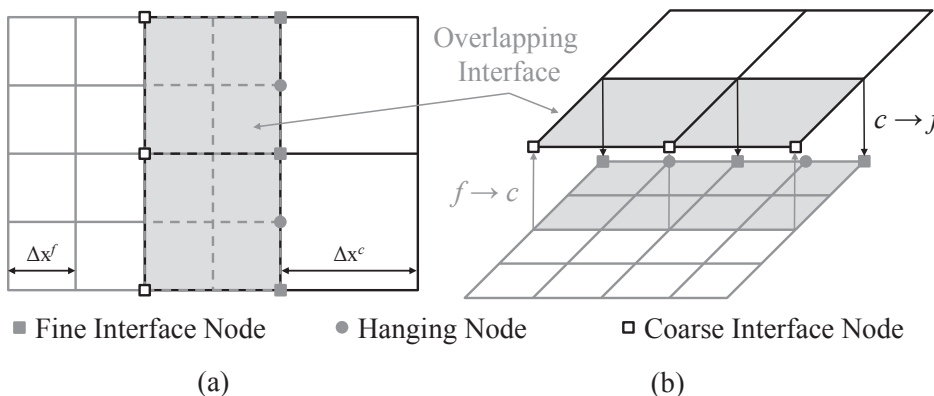


Figure 3.11: Interface Structure (2D)

and stress  $\sigma$ , do not depend on the grid level (neglecting the discretization error), this defines the conditions for transition from one grid level to another.

Secondly, we have to ensure the temporal correctness of the scheme. This means that all grid levels must advance synchronously, since LBM is a transient method. There are different ways to determine the time steps on the refined levels. In SAM-Lattice we use the convective scaling approach. Here, the speed of sound  $c_s$  and thus the molecular velocity  $\bar{\xi}$  are kept constant in the complete computational domain [35]. From the definition of the molecular velocity  $\bar{\xi} = \Delta x_n / \Delta t_n = \text{const}$ , we see directly that the ratio of the time steps on different levels is equal to grid spacing ratio, or:  $\Delta t_n = \Delta t_0 / 2^n$ . This implies that a level  $n + 1$  must do two time steps, while level  $n$  is doing one time step. The requirement of multiple time steps on finer levels per one time step on the coarse level is called nested time stepping. Another possibility for the time step determination is the diffusive scaling [100]. Here, the molecular velocity and the speed of sound are doubled when the spacing is halved. The consequence is that a level  $n + 1$  must do four time steps while level  $n$  is doing one. The diffusive scaling causes substantial higher computational costs compared to the convective scaling. Diffusive and convective scaling will be examined in more detail in the next section, dealing with accuracy issues.

There are different concepts for the definition and handling of the interface between grid levels known. These concepts can be subdivided into volumetric and node based approaches. Volumetric approaches are proposed, e.g., in [19, 49, 102], and will not be explained further, since they are not used in SAM-Lattice. In SAM-Lattice the node based approach is applied. The implementation is mainly based on the work of [25, 37, 135, 136]. Figure 3.11 depicts the interface for the highlighted area of Fig. 3.10 and will be used to schematically explain the interface definition. As first proposed by [136], the interface is not based on a line, instead it extends over one coarse cell, as it is highlighted in Fig. 3.11. In these interface cells, the parent and the child cells are retained for the calculation. In non-interface cells, the parent cells are not used during the calculation and thus removed after the refinement to save memory. For a

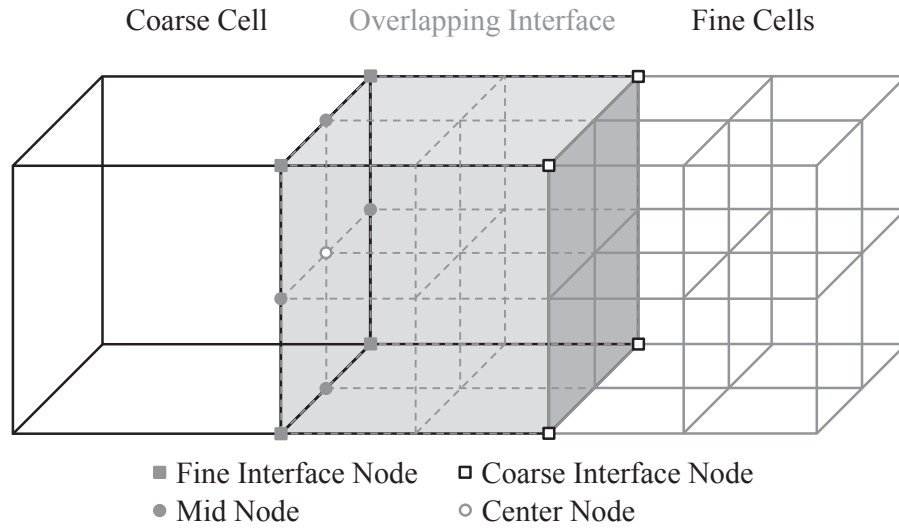


Figure 3.12: Interface Structure (3D)

better visualization of the overlapping of the coarse and fine level, the grids of the two levels are virtually decomposed in Fig. 3.11(b). In our node based LB approach, the calculation nodes reside on the corners of the depicted cells. As we can see in Fig. 3.11, the nodes of parent cells at the interface have counterparts on the fine level. This relationship is the key element of the level coupling. The coarse interface nodes are those nodes of the parent interface cells which are lacking neighbors on the coarse level and thus need a boundary condition. Analogous, the fine interface nodes are the nodes of the child interface cells, lacking neighbors on the fine level and having a counterpart on the coarse level. Through the overlapping interface, the distance between coarse and fine interface nodes is one coarse cell width. The interface nodes have a conventional node as counterpart on the neighboring level, which is called partner node. The spatial distance between the interpolation of coarse and fine interface nodes shall reduce the impact of the interpolations among each other. As said above, the boundary or grid transition condition at the interface is given by the continuity of the hydrodynamic moments. The calculation procedure for the distribution values at the interface nodes is performed in a different way for the SRT and the MRT model and will be introduced in the subsequent subsections. In Fig. 3.11(b) the coupling of the grid levels is indicated by arrows. The notation " $f \rightarrow c$ " expresses that here the information on the fine grid is used to calculate the distributions on the coarse grid, and " $c \rightarrow f$ " vice versa.

Beside the fine interface nodes, there are further nodes in the child interface cells missing neighbors on the fine level. These hanging nodes need a boundary treatment as well. Since they do not have a partner node on the coarse level, the hydrodynamic quantities or distribution values, respectively, must be obtained by interpolation from the neighboring nodes on the fine level. Yu et al. [136] proposed to use a symmetric, cubic polynomial for this interpolation pro-

cedure, to avoid spatial asymmetry of the quantities. This interpolation process involves four nodes along a line defined by the cell edge in 2D. If some interpolation partners are missing, e.g., at domain boundaries, the cubic interpolation is replaced by a lower order interpolation or extrapolation [25]. In SAM-Lattice all distribution values at the hanging nodes are interpolated in this fashion. This approach produced better results compared to only interpolating the missing distribution values.

In 3D, the interface structure is a bit more complex, see Fig. 3.12, but easy to overview with the knowledge of the 2D case. Again, the interface is an overlapping of the parent cell on the coarse level and its child cells on the fine level. The coarse and fine interface nodes result from their definition above and are handled the same way. Differences occur for the hanging nodes. The hanging nodes on the edges of the coarse interface cells are called mid nodes and are treated analog to the 2D case by a cubic interpolation along a line, which is defined by the according cell edge. For the remaining hanging nodes at the center of the cell faces, named center nodes, a bi-cubic interpolation is used [37]. The order of the interpolation is decreased for an incomplete number of interpolation nodes, i.e., stencils, analog to 2D.

### 3.7.2 SRT Scaling

This subsection introduces the calculation procedure for the interface nodes, applied with the SRT model, which is called scaling. The scaling procedure is performed after the advection and before the collision step, according to [31]. As said above, at coarse interface nodes we need to calculate the distribution values from the information at the fine partner node and vice versa for fine interface nodes. The transition condition requires that the hydrodynamic quantities density  $\rho$ , momentum  $\rho\mathbf{u}$ , and stress  $\sigma$  are equal on both levels, i.e., at the interface node and its partner node. The starting point of the scaling procedure is the decomposition of the distribution in its equilibrium and non-equilibrium part, cf. Eq. (3.20):

$$f_i = f_i^{eq} + f_i^{neq}. \quad (3.73)$$

Since the equilibrium distribution depends only on  $\rho$  and  $\mathbf{u}$ , see Eq. (3.8), the equilibrium part is equal for the nodes on both levels. (Remember that the convective scaling is applied;  $c$  indicates values on the coarse grid and  $f$  on the fine grid):

$$f_i^{eq,c} = f_i^{eq,f}. \quad (3.74)$$

For the non-equilibrium part things are different. Due to the discrete lattice effect in Eq. (3.21) the non-equilibrium must be rescaled to retain the hydrodynamic stresses when changing the



grid level. The transition condition yields for a component of the stress tensor:

$$\sigma_{\alpha\beta} = -\frac{\Omega^c \nu}{c_s^2 \Delta t^c} \sum_{i=0}^{q-1} \xi_{i,\alpha} \xi_{i,\beta} f_i^{neq,c} = -\frac{\Omega^f \nu}{c_s^2 \Delta t^f} \sum_{i=0}^{q-1} \xi_{i,\alpha} \xi_{i,\beta} f_i^{neq,f} \quad (3.75)$$

$$\Rightarrow \frac{\Omega^c \nu}{c_s^2 \Delta t^c} f_i^{neq,c} = \frac{\Omega^f \nu}{c_s^2 \Delta t^f} f_i^{neq,f} \quad \forall i \quad (3.76)$$

$$\Leftrightarrow f_i^{neq,c} = \frac{\Omega^f \Delta t^c}{\Omega^c \Delta t^f} f_i^{neq,f} = \frac{2\Omega^f}{\Omega^c} f_i^{neq,f} \quad \forall i \quad (3.77)$$

$$\Leftrightarrow f_i^{neq,f} = \frac{\Omega^c \Delta t^f}{\Omega^f \Delta t^c} f_i^{neq,c} = \frac{\Omega^c}{2\Omega^f} f_i^{neq,c} \quad \forall i. \quad (3.78)$$

For the last term in Eqs. (3.77) and (3.78), the ratio of the time steps between two subsequent grid levels, resulting from the convective scaling, was considered.

By means of Eqs. (3.74) and (3.77) or (3.78) the distribution values at the coarse or fine interface nodes are calculated from the values at their partner nodes. In SAM-Lattice all distribution values at the interface nodes are constructed by this procedure from partner nodes.

### 3.7.3 MRT Scaling

The SRT scaling procedure can in principle be applied with the MRT model, too. This effects that all non-equilibrium moments of the MRT are scaled with the same factor as the stresses. But in general, one has more possibilities for the scaling procedure of the non-equilibrium moments, since the transition conditions hold only for the hydrodynamic moments  $\rho$ ,  $\mathbf{u}$ , and  $\boldsymbol{\sigma}$ . The remaining moments can be changed between the grid levels, e.g., to increase stability. For this reason, we apply a different scaling operation in SAM-Lattice, when the MRT model is used.

Starting point for the scaling operation is the moment vector  $\mathbf{m}^p = \mathbf{M} \mathbf{f}^p$  (Eq. (3.33)) calculated at the partner node. The equality of  $\rho$  and  $\rho \mathbf{u}$  on both grid levels postulates:

$$m_0^c = m_0^f \quad \& \quad m_3^c = m_3^f \quad \& \quad m_5^c = m_5^f \quad \& \quad m_7^c = m_7^f. \quad (3.79)$$

From the relation between the hydrodynamic stresses  $\sigma_{\alpha\beta}$  and the moments  $m_i$  in Eq. (A.3), scaling relationships for the moments  $m_1, m_9, m_{11}, m_{13}, m_{14}, m_{15}$  on the coarse and the fine level can be established (see Eqs. (A.4) and (A.5)). The remaining moments  $m_2, m_4, m_6, m_8, m_{10}, m_{12}, m_{16}, m_{17}, m_{18}$  at the interface node can be chosen arbitrarily, since they do not represent hydrodynamic moments. Positive experience concerning stability has been made by simply copying the values of these moments from the partner node, i.e., no scaling of these moments. Another possibility is to set these moments to 0, the scaling reduces in this case to the regularized approach of [74]. A different approach based on a direct scaling of the moments  $\mathbf{m}^p$  was proposed by [37, 126].

Once the moment vector  $\mathbf{m}^i$  at the interface node has been calculated, the distribution values are achieved by a re-transformation  $\mathbf{f}^i = \mathbf{M}^{-1}\mathbf{m}^i$ . This mathematical operation fixes all distribution values at the interface node directly.

### 3.7.4 Nested Time Stepping

The convective as well as the diffusive scaling require nested time stepping, as explained above. Figure 3.13 depicts the nested time stepping schematically for the convective scaling, i.e., the time step is halved from level to level. To generate a valid set of distribution values at the interface nodes, the scaling procedure presented in the last subsections is needed in every time step of the according level. This means that a scaling coarse to fine is necessary for the nested time steps, too. For example, the scaling for the fine interface nodes on level 1 must be performed at time  $t_0 + \Delta t_1$ , see Fig. 3.13. Since the information of the coarse partner nodes is only available at time  $t_0$  and  $t_0 + \Delta t_0$ , the hydrodynamic quantities or distribution values must be approximated at time  $t_0 + \Delta t_1$ .

Different temporal interpolation techniques were proposed for this purpose, e.g., by Crouse [25]. An upwind interpolation is sufficient for stationary flows. Here the values at time  $t_0 + \Delta t_0$  are used for scaling of the fine interface nodes at  $t_0 + \Delta t_1$ . At least a linear interpolation is required for transient flows. The values of the coarse partner nodes at time  $t_0 + \Delta t_1$  are then obtained as means of the values at time  $t_0$  and  $t_0 + \Delta t_0$ . In addition, quadratic interpolation in time is proposed in literature. According to [25], the quadratic time interpolation shows only small improvements compared to the linear one. But it requires the saving of a third value of the quantity to be interpolated in time. Due to these facts, a linear interpolation in time is used for the nested time stepping in SAM-Lattice.

## 3.8 Accuracy and Stability

In the previous sections different issues concerning the accuracy and stability of the Lattice Boltzmann Method have been addressed. In this section a more general, conclusive description of these topics will be provided, since accuracy and stability play an important role for theoretical considerations and practical applications.

### 3.8.1 Accuracy Issues

The Lattice Boltzmann equation is derived by a second order accurate quadrature rule (Eq. (3.12)) in combination with a coupling of the space and time step from the discrete Boltzmann equation, which recovers the Navier-Stokes equations with second order accuracy in the Knudsen number  $\epsilon$ . Thus, LBE results in a formally second order accurate scheme in space and time for the compressible Navier-Stokes equations. The overall error of the scheme  $E$  contains the truncation errors  $E_t = O(\Delta t^2)$  and  $E_x = O(\Delta x^2)$  and an error in the Knudsen number  $E_\epsilon = O(\epsilon^2)$ .

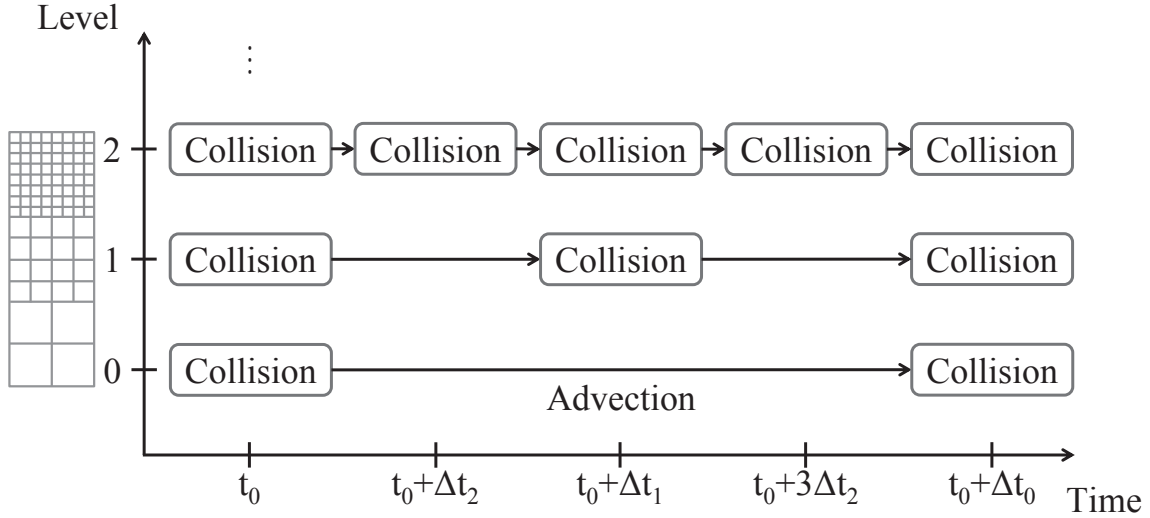


Figure 3.13: Nested Time Stepping

The Knudsen number can be interpreted as  $\epsilon = \Delta x/L$  in the LBM context [50], where  $L$  is a characteristic length of the flow, and hence the Knudsen error reduces to  $E_\epsilon = \mathcal{O}(\Delta x^2)$ . Consequently, the overall error is expressed by:

$$E = E_t + E_x = \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2). \quad (3.80)$$

The Lattice Boltzmann Method is a compressible scheme as the density is not a constant quantity and is proportional to the pressure, cf. Eq. (3.7). As introduced in this thesis, the LBM is used for flows at small Mach numbers, i.e.,  $Ma < 0.3$ , which are commonly regarded as incompressible. Nevertheless, an additional modeling (compressibility) error  $E_{Ma}$  is introduced, if an incompressible flow is modeled with LBM [46, 56, 74]. It can be shown [22, 74] that the compressibility error is of the order  $E_{Ma} = \mathcal{O}\left(\frac{\Delta t^2}{\Delta x^2}\right)$ . The overall error  $E$  of the scheme for incompressible simulations results in:

$$E = E_t + E_x + E_{Ma} = \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2) + \mathcal{O}\left(\frac{\Delta t^2}{\Delta x^2}\right). \quad (3.81)$$

From Eq. (3.81) we see that the compressibility error produces a coupling between the discrete space and time steps  $\Delta x$  and  $\Delta t$ . To obtain a scheme that is second order accurate in space,  $E_{Ma}$  must reduce to  $\mathcal{O}(\Delta x^2)$ . This behavior can be achieved by choosing  $\Delta t \propto \Delta x^2$ , what is called diffusive scaling. An effect of the diffusive scaling, which has already been shortly addressed, is the change of the speed of sound  $c_s$  in the lattice. In the diffusive scaling  $c_s$  is doubled, when  $\Delta x$  is halved. That means, the Mach number  $Ma$  is halved, too. The relaxation parameter  $\Omega$  is kept constant when the diffusive scaling is applied, since  $\Omega \propto \frac{\Delta x^2}{\Delta t}$ .

But, the diffusive scaling has an impact on the temporal accuracy as well. We consider the

following: The spacing  $\Delta x$  is halved between two grids and thus the error is quartered for a second order accurate scheme. According to the diffusive scaling,  $\Delta t$  must be divided by four, when  $\Delta x$  is divided by two. Thus, the scheme is only first order accurate in time for incompressible flows, when using diffusive scaling, what is formally shown by [56, 63]. On the other hand Holdych et al. [56] show that for a fixed grid spacing  $\Delta x$ , LBM is second order accurate in time for incompressible flows.

In the convective scaling, we use  $\Delta t \propto \Delta x$  what yields to constant speed of sound  $c_s$  on different grids and thus identical  $Ma$ . Due to  $c_s = const$ , the convective scaling is often called acoustic scaling. The compressibility error in incompressible flows then reduces to  $E_{Ma} = \mathcal{O}(1)$  and is thus in general non-convergent [56]. Nevertheless, the overall error  $E$  will decrease quadratically with  $\Delta x$  until the compressibility error is getting dominant [56, 74].

The effects of convective scaling have consequences on the grid refinement technique proposed in the previous section. As convective scaling is used between the grid levels, we have to ensure that the second order convergence of the scheme is preserved. This can be achieved by staying in the over relaxation region ( $\Omega > 1$ ) in all levels, as proposed by [35]. In the over relaxation region, the compressibility error is not dominating [56] and a second order behavior of the overall error occurs. This can be seen from the numerical results of [49, 71].

Besides the omnipresent truncation errors  $E_t$  and  $E_x$  of the method and the compressibility error  $E_{Ma}$ , which is only present in incompressible flow simulations, further errors may be added to the overall error  $E$ . Another error is introduced by insufficient boundary conditions  $E_{BC}$ . In Section 3.5, we classified boundary conditions as first and second order accurate boundary conditions. In the notation used in this section this means  $E_{BC} = \mathcal{O}(\Delta x)$  for first order and  $E_{BC} = \mathcal{O}(\Delta x^2)$  for second order accurate conditions. In the case of first order boundary conditions, the global overall error  $E$  will be  $\mathcal{O}(\Delta x)$ . However, some authors, e.g., [35, 57] argue that locally seen the solution accuracy is only decreased to first order near the boundary. A reversing of this statement is that a local quantity, which is some distance away from the boundary, can still converge with second order accuracy in this case.

A further error  $E_{IC}$ , generated by insufficient initial conditions, may be addressed shortly. This error is only of importance for transient flows, since the solutions of steady flows are independent of the initial conditions. However, for unsteady flows, inconsistent initial conditions can create so called initial layers, which induce large errors  $E_{IC}$  [15, 46]. Nevertheless, a dependence of  $E_{IC}$  in terms of  $\Delta x$  and  $\Delta t$  is not known.

### 3.8.2 Stability Issues

From a numerical point of view, the Lattice Boltzmann Method is an explicit finite difference scheme for the computation of the distribution values  $f_i$  [114]. As other explicit schemes, LBM is not unconditionally stable. Sterling and Chen [114] showed by a von Neumann stability analysis that SRT-LBM is getting unstable for  $\Omega \rightarrow 2$ . From Eq. (2.37) we see that the case  $\Omega = 2$  corresponds to zero viscosity and thus infinite Reynolds number. The instability manifests by negative distribution values  $f_i$ , which are physically meaningless since the values  $f_i$  represent

densities. Another result of [114] depicts that the stability of the scheme depends on the flow velocity. Even for small  $\Omega$  instabilities occur, if the velocity is high in proportion to  $c_s$ , i.e., if the low Mach number assumption is violated. In this case, the nonlinear terms in the equilibrium distribution (Eq. (3.8)) are getting dominant. The source of the instability is seen by Sterling and Chen [114] in the missing H-theorem of the discrete LB model. To overcome this drawback, a class of entropic Lattice Boltzmann methods has been developed, see, e.g., [6], which provide a H-theorem to obtain higher numerical stability. Besides the entropic LBM, other SRT based schemes have been developed to improve the stability of the method.

Li et al. [77] propose to modify the relaxation rate to ensure numerical stability, i.e., the positivity of the distributions  $f_i$ :

$$\tilde{\Omega} = \min_i \left( \Omega, \frac{f_i}{f_i - f_i^{eq}} \right). \quad (3.82)$$

This procedure is also known as positivity rule or fix-up technique [127]. Disadvantages of the fix-up technique are that by using  $\tilde{\Omega}$ , additional or numerical viscosity, respectively, is introduced and that this scheme does not suppress another numerical artifact called spurious oscillations, which are in fact numerical waves [12]. The numerical viscosity of the fix-up technique is especially crucial when a low dissipative method is needed, like in aero-acoustics or large eddy simulations. Nevertheless, the fix-up rule is used in SAM-Lattice with the SRT model as a salvation rule in turbulent calculations.

Brownlee et al. [12] propose to use an Ehrenfest step if distributions are getting negative. In this technique, the distributions at a node are set to their equilibrium values if the collision process would generate non positive distributions. Sending the distributions to equilibrium corresponds to a relaxation rate  $\Omega = 1$  and thus adds numerical viscosity. Further techniques based on filtering operations have been proposed [13, 101], but have not yet been tested in SAM-Lattice. The implementation and evaluation of such schemes is left for future work.

Stability analyses have been performed for the multiple relaxation time LBM, too [72, 101]. MRT-LBM also holds numerical instability for  $\Omega \rightarrow 2$ , but offers the possibility to choose different relaxation rates for non-hydrodynamic moments, often called kinetic or ghost modes. In this way the stability of the scheme can be dramatically improved [29, 79] without adding numerical shear viscosity. Instead, additional dissipation is added to the non-hydrodynamic, kinetic modes. A positive feature of MRT is that by adjusting the bulk viscosity, spurious oscillations can be damped [26, 101]. On the other hand, the additional dissipation of the kinetic modes can be disadvantageous when dealing with aero-acoustic simulations [83], what shows that, depending on the physics to be modeled, a compromise between stability and accuracy must be made.

The considerations on stability made so far, do not include features like boundary conditions and grid refinement. Imperfect boundary conditions may not only reduce the accuracy, but also give rise to instabilities. For instance, [84] investigate numerically the stability of interpolation based bounce back schemes, as a function of  $\Omega$  and the boundary position  $q$ . In general, boundary conditions reduce the stability of the scheme, since they introduce additional disturbances.

But, a universal relationship between boundary treatment and stability can hardly be made, because it depends on different parameters like flow geometry and flow conditions.

Similar effects can be observed for grid refinement. The transition from a grid level to another one introduces disturbances, which promote instabilities. Besides the author's experience of this mechanism, [101] report similar observations. Different numerical tests performed by the author revealed that the multi-level MRT scheme in SAM-Lattice is stable, as long as the relaxation parameter in the coarsest level is restricted to  $\Omega_0 < 1.995$ . This value should be seen as a guide value, not as a hard limit, since the exact stability limit is again depending on the flow configuration and flow conditions. As the multi-level SRT scheme is less stable than the MRT version, no special investigations have been made for SRT. Lagrava et al. [71] suggest to use a filtering operation for the scaling operation of the coarse interface nodes and tested this procedure in 2D. A box filter is used for filtering the non-equilibrium part on the fine level before scaling to the coarse level. This procedure has been implemented and tested by the author. Nevertheless, it did not improve stability in 3D remarkably. Promising results regarding the stability on refined grids have been published very recently by Touil et al. [128]. They use a similar filtering operation for mapping from fine to coarse in combination with a selective filtering in the entire domain, as proposed by [101]. Due to the novelty of this paper, it has not yet been tested. This will be done in future work.

---

## Large Eddy Simulation for Lattice Boltzmann Methods

---

Most flows of technical interest are of turbulent nature. To accurately calculate turbulent flows with the LBM schemes introduced so far, all structures of the flow need to be resolved spatially and temporally. In the context of turbulent simulations this is called *direct numerical simulation* (DNS). The big drawback of DNS is its tremendous computational demand, due to the high resolution. It can be shown that for homogeneous and isotropic turbulent flows, which are considered as the simplest turbulent flows, the computational demand is of  $O(Re^3)$  [104]. For typical technical flows, where Reynolds numbers  $Re \sim 10^5 - 10^8$  are not uncommon, a DNS requires computational resources that are not available by today's super computers and will not be available in foreseeable future. Direct numerical simulation is thus restricted to basic research for simple flows at low Reynolds numbers.

Different techniques and models have been developed to approximate turbulent simulations at lower computational costs, see, e.g., [34, 104] for an overview. One such technique is the *large eddy simulation* (LES). In this chapter a consistent and complete LES module for LBM is proposed, which is applied in SAM-Lattice.

### 4.1 Principles of Large Eddy Simulation

The main idea of LES is the scale separation of the flow, i.e., the separation into large and small scales. The large scale structures, i.e., large eddies, are then computed directly, whereas the small scale structures, that would need a high spatial and temporal resolution, are considered by a model. Figure 4.1 depicts this schematically. On the left hand side (Fig. 4.1(a)) the separation in physical space is shown. The separation process depends on the cutoff length, which is commonly associated to the grid spacing  $\Delta x$ , and thus the small scales are often referred to

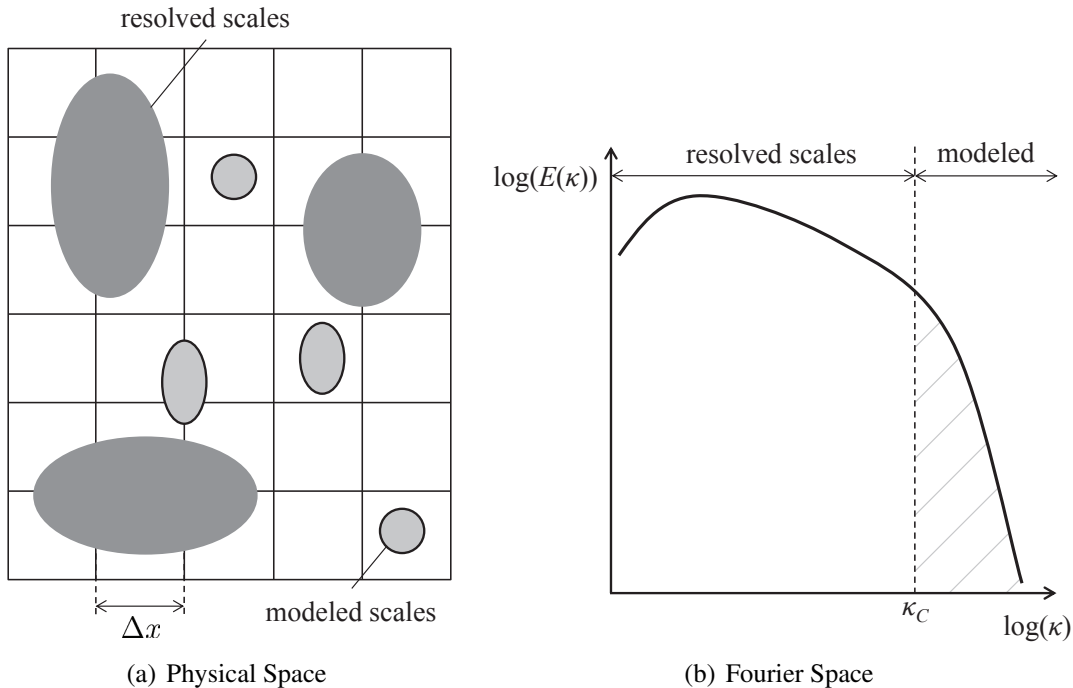


Figure 4.1: Scale Separation (based on [104])

as subgrid scales. Figure 4.1(b) shows the scale separation in Fourier space, by means of wave number  $\kappa$  and turbulent kinetic energy  $E(\kappa)$ . Here we see that the large scales with wave numbers  $\kappa < \kappa_c$  are calculated, i.e. resolved, and the remaining part, which contains less energy, is modeled. The cutoff wave number  $\kappa_c$  is related to the cutoff length in physical space, see [104] for details. The LES procedure can be physically justified by different facts [39]: The large eddies contain most of the turbulent energy and they depend strongly on the flow configuration. Furthermore, large eddies are characterized to be inhomogeneous, anisotropic, long-lasting and diffusive. All these properties make the large eddies hard to model. The small eddies, in contrary, contain only a minor part of the turbulent kinetic energy, are more homogeneous and isotropic, relatively short-dated and dissipative, what makes a modeling easier. To get accurate, reliable results, it is commonly recommended to choose the cutoff length in a way to resolve at least 80% of the turbulent kinetic energy [38, 94]. By this technique, the computational demand for calculating turbulent flows is dramatically reduced.

Mathematically, the scale separation is realized by a filtering operation, which is represented as a convolution product here; for other definitions see [104]. In physical space this operation correlates to a high-pass filter since the small scales are to be removed. This corresponds to a low-pass filter in Fourier space for removing the high frequencies or wave numbers. In what follows, only the physical space is considered. A quantity  $\phi(\mathbf{x}, t)$  is decomposed into a resolved



part  $\bar{\phi}(\mathbf{x}, t)$  and an unresolved part  $\phi'(\mathbf{x}, t)$ .

$$\phi(\mathbf{x}, t) = \bar{\phi}(\mathbf{x}, t) + \phi'(\mathbf{x}, t) \quad (4.1)$$

The resolved or filtered part is defined by the convolution product [39]:

$$\bar{\phi}(\mathbf{x}, t) = \int_{-\infty}^{\infty} G(\mathbf{x} - \mathbf{x}')\phi(\mathbf{x}', t)d\mathbf{x}' = G * \phi. \quad (4.2)$$

The convolution kernel  $G$  is a characteristic for the applied filter, which is linked to the cutoff length in physical space  $\bar{\Delta}_c$ . For convenience, the right hand side of Eq. (4.2) introduces a short notation of the filtering operation. The filter kernel  $G$  in  $\mathbb{R}^3$  is obtained by tensorizing mono-dimensional kernels [104]:

$$G(\mathbf{x} - \mathbf{x}') = \prod_{i=0}^3 G_i(x_i - x'_i). \quad (4.3)$$

Common choices for the mono-dimensional kernel  $G_i$  are the box filter, the Gaussian filter or the Fourier cutoff filter. The box filter is defined by (see [39, 104] for the remaining ones):

$$G_i(x_i - x'_i) = \begin{cases} \frac{1}{\bar{\Delta}_c} & \text{if } |x_i - x'_i| \leq \frac{\bar{\Delta}_c}{2} \\ 0 & \text{else.} \end{cases} \quad (4.4)$$

The box filter is visualized in Fig. 4.2 for  $\bar{\Delta}_c = 1$ . We see that only values in the interval  $(x_i - x'_i) \pm \bar{\Delta}_c/2$  are not equal 0 and are thus incorporated in the averaging process. Commonly the filter width or cutoff length  $\bar{\Delta}_c$  is identified with the grid spacing  $\Delta x$  [39].

The next step to obtain a large eddy scheme is the filtering of the governing analytical equations, e.g., the Boltzmann equation or the Navier-Stokes equations. Through the filtering process some quantities in the equations occur, which are not directly computable and thus need modeling by a so called subgrid scale model. This will be shown in the next subsection. Before, some considerations on the effective filter of the numerical scheme are in order.

The filtering process introduced so far is applied to the analytical equations and thus often called analytical filter. Since our aim is to solve the filtered equations with a numerical method further effects occur, which can be related to a filtering operation [41]. Firstly, the discrete computational grid acts as a filter as well. As a consequence, no scales smaller than the grid spacing can be resolved, what is a reason for the common choice  $\bar{\Delta}_c = \Delta x$ . In the frequency domain this restriction from the grid leads to a maximum resolvable frequency, which is given by the Nyquist-Shannon theorem. Secondly, the errors introduced by the numerical approximation of derivatives can be interpreted as a numerical filter applied to the exact solution. Thirdly, the modeling errors introduced by some subgrid scale model can also be interpreted as a numerical filter. All these filtering operations combined lead to the effective filter of the numerical solution. Since the determination of the effective filter and the associated effective cutoff length is

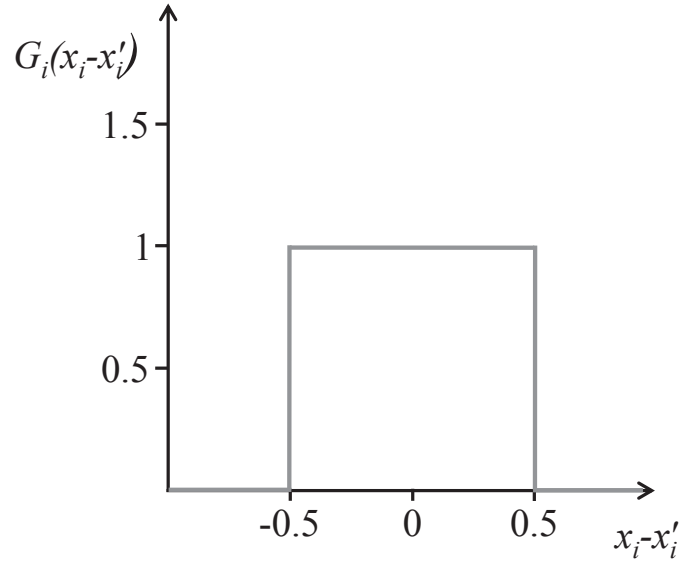


Figure 4.2: Box Filter ( $\overline{\Delta_c} = 1$ )

mathematically a very difficult problem, usually only the cutoff length  $\overline{\Delta_c} = \Delta x$  of the analytical filter is considered in practice. The other sources remain disregarded, see [104] for more details.

The large eddy approach has some demands on the numerical method used for solving the filtered equations. On the one hand, the numerical method must be transient due to the intrinsic time dependence of turbulence. On the other hand, the numerical scheme must be low dissipative, as numerical dissipation tends to remove energy from the highest resolved wave numbers [34, 92]. LBM fulfills both criteria, it is a time dependent scheme and known to be low dissipative, as shown, e.g., by Marié et al. [83]. So, at a first glance the Lattice Boltzmann Method is well suited for large eddy simulation.

## 4.2 Application to the Lattice Boltzmann Method

Hou et al. [57] proposed first the use of a large eddy model within the Lattice Boltzmann Method. They applied the filtering operation of Eq. (4.2) to the discrete velocity Boltzmann equation (Eq. (3.9)) to obtain a LES scheme. The derivation of LES-LBM presented in the following is mainly based on the subsequent, more consistent work of [81, 105].

### 4.2.1 Filtered Lattice Boltzmann Equation

By applying the filtering operation of Eq. (4.2) to the Boltzmann equation (Eq. (2.11)) the filtered Boltzmann equation is obtained (note:  $\bar{f} = G * f$ ).

$$\frac{\partial \bar{f}}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial \bar{f}}{\partial \mathbf{x}} = \overline{\Omega(f)} \quad (4.5)$$

For simplicity, the force term of the Boltzmann equation has been omitted in Eq. (4.5). The term on the right hand side  $\overline{\Omega(f)}$  expresses the filtering operation applied to an arbitrarily collision operator. For convenience the BGK or SRT model, respectively, is used afterwards. Filtering the BGK model (Eq. (2.19)) results in:

$$\overline{\Omega(f)} = G * \left[ -\frac{1}{\tau} (f - f^{eq}) \right] = -\frac{1}{\tau} (\bar{f} - \overline{f^{eq}}). \quad (4.6)$$

The problem in evaluating  $\overline{\Omega(f)}$  is that for the calculation of  $\overline{f^{eq}}$  (Eq. (2.15)) the unfiltered quantities  $\rho = \bar{\rho} + \rho'$  and  $\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}'$  are needed. In general, these quantities are not computable, since the unresolved parts  $\rho'$  and  $\mathbf{u}'$  are unknown. As proposed by [57] the filtered equilibrium distribution  $\overline{f^{eq}}$  can be replaced by the equilibrium distribution build with the filtered macroscopic values  $f^{eq}(\bar{\rho}, \bar{\mathbf{u}})$  and a correction term  $R$ .

$$\begin{aligned} \overline{\Omega(f)} &= \Omega(\bar{f}) + R = -\frac{1}{\tau} [\bar{f} - f^{eq}(\bar{\rho}, \bar{\mathbf{u}})] + R \\ R &= \frac{1}{\tau} [\overline{f^{eq}} - f^{eq}(\bar{\rho}, \bar{\mathbf{u}})] \end{aligned} \quad (4.7)$$

The correction term is necessary due to the lack of commutativity of the filtering operator and the nonlinearity in  $f^{eq}(\rho, \mathbf{u})$ . Missing commutativity is shown by, e.g.,  $\overline{uv} \neq \bar{u} \bar{v}$ , see [39, 41, 104] for further mathematical properties of the filtering operator. The filtered macroscopic quantities as  $\bar{\rho}, \bar{\mathbf{u}}$  are build according to their definition in Eqs. (2.4) et seq., but with the filtered distribution values  $\bar{f}$  instead of  $f$ . From a physical point of view, the correction term  $R$  includes unresolved subgrid quantities and thus has to be obtained by a model.

Finally, the filtered Boltzmann-BGK equation reads as:

$$\frac{\partial \bar{f}}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial \bar{f}}{\partial \mathbf{x}} = -\frac{1}{\tau} [\bar{f} - f^{eq}(\bar{\rho}, \bar{\mathbf{u}})] + R. \quad (4.8)$$

The discrete filtered Lattice Boltzmann equation is derived from Eq. (4.8) by first discretizing in velocity space, as shown in Subsection 3.1.1, and a subsequent discretization in space and time, as shown in Subsection 3.1.2, what yields to Eq. (4.9). This discretization process is shown completely in [81].

$$\bar{f}_i(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) = \bar{f}_i(\mathbf{x}, t) + \Omega \left[ f_i^{eq}(\bar{\rho}(\mathbf{x}, t), \bar{\mathbf{u}}(\mathbf{x}, t)) - \bar{f}_i(\mathbf{x}, t) \right] + R_i \quad (4.9)$$

Compared to the standard, unfiltered Lattice Boltzmann equation (Eq. (3.16)) the filtered version in Eq. (4.9) contains additionally the subgrid terms  $R_i$ . The discrete equilibrium distribution  $f_i^{eq}$  is identical to Eq. (3.8). But, it is build, as in the continuous filtered case, with  $\bar{\rho}$  and  $\bar{\mathbf{u}}$ . For the treatment of the subgrid terms  $R_i$  different ways are known. In SAM-Lattice, we use the eddy viscosity assumption, which is also called Boussinesq assumption, for modeling of these subgrid terms. This modeling approach is described by Sagaut [104] with the following hypothesis:

*“The energy transfer mechanism from the resolved to the subgrid scales is analogous to the molecular mechanisms represented by the diffusion term, in which the viscosity appears.”*

In conclusion to this, the transfer of turbulent energy to subgrid scales is mathematically modeled in the same way as momentum transfer, i.e., the kinematic viscosity  $\nu$  in the governing equations is replaced by an effective kinematic viscosity  $\nu_{eff}$ . Besides the molecular viscosity  $\nu$ , the effective viscosity contains a turbulent contribution, the eddy viscosity  $\nu_t$ .

Hou et al. [57] applied the eddy viscosity assumption to LBE by simply introducing the effective viscosity  $\nu_{eff}$  in the relaxation parameter  $\tau$ . They pointed out that the filtered Navier-Stokes equation, also based on the eddy viscosity assumption, can be derived from this model. Recently, [81] proofed this formally by a Chapman-Enskog expansion for the continuous, filtered Boltzmann-BGK equation in the incompressible limit. Under these assumptions, the correction term  $R$  can be incorporated through an effective relaxation parameter in the BGK model:  $\tau_{eff} = \tau + \tau_t$ .

With the eddy viscosity assumption the discrete filtered LBE results in:

$$\bar{f}_i(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) = \bar{f}_i(\mathbf{x}, t) + \Omega_{eff} \left[ f_i^{eq}(\bar{\rho}(\mathbf{x}, t), \bar{\mathbf{u}}(\mathbf{x}, t)) - \bar{f}_i(\mathbf{x}, t) \right] \quad (4.10)$$

$$\Omega_{eff} = \frac{c_s^2 \Delta t}{\nu + \nu_t + 0.5 c_s^2 \Delta t}. \quad (4.11)$$

The effective relaxation parameter  $\Omega_{eff}$  includes the eddy viscosity contribution  $\nu_t$ , which must be determined by a model. Such subgrid scale models are subject of the next subsection.

Large eddy schemes based on the eddy viscosity approach can be easily implemented in a similar way for the MRT collision operator, as proposed by [67]. The outcome of this procedure is that the relaxation rate  $\Omega$  for the moments related to the momentum fluxes (see Eq. (3.35)) is replaced by Eq. (4.11).

For the sake of completeness, a different way to represent the subgrid terms  $R_i$ , which is not based on the eddy viscosity approach will be mentioned here. The basic idea of the Approximate Deconvolution Method (ADM) [115] is to calculate or at least approximate the unfiltered quantities by inverting the filtering operation [104]. By means of the unfiltered quantities density and velocity, which are received this way, the terms  $R_i$  can be determined. Recently, ADM has been adopted to LBM, for further details see [80, 105].

## 4.2.2 Subgrid Scale Models

The remaining element in the large eddy scheme is the subgrid model used to evaluate the eddy viscosity  $\nu_t$ , which is needed for the effective relaxation parameter in Eq. (4.11). As already said, the subgrid model accounts for the energy transfer from large resolved scales to subgrid scales. The most popular subgrid scale model, which is used in SAM-Lattice as well, is the Smagorinsky model [113]. In the Smagorinsky model the eddy viscosity is calculated by:

$$\nu_t = (C_s \overline{\Delta_c})^2 |\overline{\mathbf{S}}| = (C_s \Delta x)^2 |\overline{\mathbf{S}}|. \quad (4.12)$$

The eddy viscosity is depending on the dimensionless Smagorinsky constant  $C_s$ , the cutoff length of the filter  $\overline{\Delta_c}$ , which is set equal to the grid spacing  $\Delta x$ , and the norm of the resolved strain rate tensor  $|\overline{\mathbf{S}}|$ . The Smagorinsky constant is commonly chosen between 0.1–0.2, whereas it can be theoretically shown that  $C_s \sim 0.18$  for homogeneous isotropic turbulence [41]. In practice,  $C_s$  is used as a calibration parameter of the model.

For the solution of Eq. (4.12), the norm of the resolved strain rate tensor needs to be calculated. The definition of  $\overline{\mathbf{S}}$  and the applied norm are:

$$\overline{S}_{\alpha\beta} = \frac{1}{2} \left( \frac{\partial \overline{u}_\alpha}{\partial x_\beta} + \frac{\partial \overline{u}_\beta}{\partial x_\alpha} \right) \quad (4.13)$$

$$|\overline{\mathbf{S}}| = \sqrt{2 \overline{S}_{\alpha\beta} \overline{S}_{\alpha\beta}}. \quad (4.14)$$

The strain tensor  $\overline{\mathbf{S}}$  can be calculated from a finite difference stencil, or directly from the values of the distribution function  $\overline{f}_i$ , what is second order accurate as shown by [68]. To realize this we use the stress-strain relationship for incompressible flows:

$$\overline{\boldsymbol{\sigma}} = 2\mu_{eff} \overline{\mathbf{S}} = 2\rho\nu_{eff} \overline{\mathbf{S}}. \quad (4.15)$$

Calculation of the filtered stress tensor  $\overline{\boldsymbol{\sigma}}$  from the distribution function is performed with help of Eq. (3.21). For convenience some short notation is introduced:

$$\mathbf{\Pi}_{discr.}^{neq} := \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i [\overline{f}_i - f_i^{eq}(\overline{\rho}, \overline{\mathbf{u}})] \quad (4.16)$$

$$\Rightarrow \overline{\boldsymbol{\sigma}} = -\frac{\Omega_{eff} \nu_{eff}}{c_s^2 \Delta t} \mathbf{\Pi}_{discr.}^{neq}. \quad (4.17)$$

From Eqs. (4.15) and (4.17) a relation for the strain rate  $\overline{\mathbf{S}}$  in dependence on the distribution values  $\overline{f}_i$  is achieved:

$$\overline{\mathbf{S}} = -\frac{\Omega_{eff}}{2\rho c_s^2 \Delta t} \mathbf{\Pi}_{discr.}^{neq}. \quad (4.18)$$

By inserting this in the Smagorinsky model we obtain (please note the difference between the Smagorinsky constant  $C_s$  and the lattice speed of sound  $c_s$ ):

$$\nu_t = (C_s \Delta x)^2 \frac{\Omega_{eff}}{2\rho c_s^2 \Delta t} |\mathbf{\Pi}_{discr.}^{neq}|. \quad (4.19)$$

If we bear in mind that the effective relaxation rate depends on  $\nu_t$  itself (Eq. (4.11)), Eq. (4.19) ends up in a quadratic equation for  $\nu_t$ , which has only one physical meaningful, i.e., positive solution:

$$\nu_t = \frac{\sqrt{\left(\nu + \frac{1}{2}c_s^2\Delta t\right)^2 + \frac{2}{\rho}(C_s\Delta x)^2|\mathbf{\Pi}_{discr.}^{neq}|} - \left(\nu + \frac{1}{2}c_s^2\Delta t\right)}{2}. \quad (4.20)$$

Using this equation,  $\nu_t$  can be calculated with the Smagorinsky model locally at every node. This has a superior computational performance compared to a finite difference rule and is another advantage of LBM in the LES context.

Since Eq. (4.12) is just a model, a critical discussion of the limits is in order. As the dependencies show, the Smagorinsky model is based on the resolved scales, more precisely on the gradient of the resolved velocity field. So,  $\nu_t$  will always be greater zero if a spatial gradient is present in the flow, even in laminar flows [94, 104]. The Smagorinsky model can thus not calculate laminar or transitional flows and should only be applied for fully turbulent flows. Another drawback is its too dissipative behavior near walls in wall bounded flows, where the turbulence is neither homogeneous nor isotropic [76]. To overcome this deficiency, the Smagorinsky constant  $C_s$  is multiplied with a damping function according to van Driest [34, 104]:

$$C_s^{vD} = C_s [1 - \exp(-y^+/A^+)]^2 \quad A^+ = 25 \quad (4.21)$$

$$y^+ = \frac{yu_\tau}{\nu} \quad (4.22)$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}. \quad (4.23)$$

The distance to a wall is measured by the dimensionless wall distance  $y^+$ , which depends on the distance to the wall  $y$ , the friction velocity  $u_\tau$  and the kinematic viscosity  $\nu$ . The friction velocity itself is related to the wall shear stress  $\tau_w$ .  $C_s$  is reduced exponentially with  $y^+$  when reaching the wall. By using  $C_s^{vD}$  as constant in the Smagorinsky model (Eq. (4.12)) the near wall behavior is improved. Furthermore, the eddy viscosity should reduce with  $\nu_t \sim y^{+3}$  for decreasing distance to the wall  $y \rightarrow 0$  [34, 104]. To reproduce this behavior the van-Driest damping function can be slightly modified [104]:

$$C_s^{vD*} = C_s [1 - \exp(-y^+/A^+)]^{1/2} \quad A^+ = 25. \quad (4.24)$$

Besides the above introduced Smagorinsky model, different subgrid scale models have been proposed in the LBM context. Dong et al. [30] suggested the use of an inertial-range (IR) consistent Smagorinsky model in LBM. This model is very similar to the standard Smagorinsky

model, apart from the fact that a different effective viscosity definition is used in [30]. In contrast to the standard Smagorinsky model ( $\nu_{eff} = \nu + \nu_t$ ), in the IR Smagorinsky model  $\nu_{eff} = \sqrt{\nu^2 + \nu_t^2}$  is used, where  $\nu_t$  was determined in [30] by Eq. (4.12) with  $C_s = 0.18$ . Premnath et al. [96] adopted the dynamic Smagorinsky model into LBM. In the dynamic Smagorinsky model, Eq. (4.12) is still applied, but the constant  $C_s = C_s(\mathbf{x}, t)$  is space and time depending.  $C_s(\mathbf{x}, t)$  is calculated by the Germano-Lilly procedure, for details see [104]. Due to this procedure, the dynamic model is feasible for the calculation of transitional flows, since the constant  $C_s$  vanishes in laminar flow. Furthermore, this model produces the desired  $y^{+3}$  decline in the near wall region [34]. A more recently developed subgrid scale model with dynamic properties is the shear improved Smagorinsky model proposed by [76] and used in LBM by [128]. This model is also designed to reproduce transitional and near wall flows correctly, but is computationally less demanding. The implementation of a dynamic model into SAM-Lattice will be part of future work.

### 4.3 Boundary Conditions for Large Eddy Simulation

The treatment of the boundary conditions in large eddy simulation needs some additional considerations. First of all, the interaction between the filter and boundary conditions is addressed. The isotropic filtering operation defined by Eq. (4.2) is strictly seen only applicable to unbounded domains and was used to present the principles of LES. For bounded domains, the filtering process introduces additional commutation errors arising from the interaction with the boundary conditions; see [104] for a detailed presentation of this topic. In the so called classic approach, which is adopted here, these additional errors are neglected and thus the filtered equations on bounded domains yield the ones presented in the last section (e.g. Eq. (4.10)). The interested reader is again referred to Sagaut [104] for a detailed motivation of this approach and a treatment of the additional commutation errors, if these are not neglected.

In the course of this section, methods for the creation of suitable boundary conditions in LES-LBM will be developed. Since there is in general no special treatment for outlet boundaries in LES, this boundary type can be handled with the schemes of Section 3.5 and will not be explicated any further here.

#### 4.3.1 Wall Boundary Conditions

For a better understanding of the demands on wall conditions in LES, the structure of turbulent boundary layers will be shortly introduced in the case of canonical boundary layers. The main characteristics of canonical turbulent boundary layers are the absence of a pressure gradient in streamwise direction, the two-dimensionality of the boundary layer in the mean, fully developed turbulence, and that there is no wall curvature or surface roughness. The line plot in Fig. 4.3 shows the mean velocity profile as a result of an accurate DNS calculation of turbulent channel flow [58] in terms of the dimensionless wall distance  $y^+$  (Eq. (4.22)) and a reduced mean

streamwise velocity:

$$u^+ = \frac{\langle u \rangle}{u_\tau}. \quad (4.25)$$

Additionally, Fig. 4.4 presents the development of different stress parts in the boundary layer versus  $y^+$  (calculated from the data of [58]). Besides the viscous stress  $\tau_v = \mu d\langle u \rangle / dy$ , the so called Reynolds stress  $\tau_R = -\rho u'v'$  is shown, where  $u'$  and  $v'$  are the turbulent velocity fluctuations in  $x$  and  $y$  direction, respectively. In this context, the mean velocity  $\langle u \rangle$  and the fluctuations, e.g.,  $u'$ , are to be understood as a Reynolds decomposition. They should not be confused with the decomposition in the context of the filtering operation (Eq. (4.1)):

$$\langle u(\mathbf{x}) \rangle = \frac{1}{T} \int_t^{t+T} u(\mathbf{x}, t') dt' \quad (4.26)$$

$$u'(\mathbf{x}, t) = u(\mathbf{x}, t) - \langle u(\mathbf{x}) \rangle. \quad (4.27)$$

More details on Reynolds decomposition can be found in Pope [94].

The Reynolds stress terms actually stem from the decomposition of the convective terms in the Navier-Stokes equations. Strictly seen, they are momentum terms, not stresses. Since these terms transfer momentum of the fluctuations and are accountable for the increased flow losses, they are interpreted analog to the viscous stresses, which transfer momentum at molecular level. The sum of these both stresses is depicted by  $\tau_{total}$ , all stresses are normalized by means of the wall shear stress  $\tau_w$  (Eq. (3.57)).

On the basis of Figs. 4.3 and 4.4 the velocity profile is subdivided into different regions and layers. The inner region, in which viscosity effects are of importance, contains three layers:

- The viscous layer which is commonly defined by  $y^+ \leq 5$ . In the viscous layer the viscous shear stress is dominant and the Reynolds stresses can be neglected, see Fig. 4.4. In combination with the assumption that  $\tau_{total}$  is constant, the following relation for the mean flow velocity can be deduced [94]:

$$u^+ = y^+. \quad (4.28)$$

Equation (4.28) is called lin law in the remaining part of this thesis.

- The buffer layer:  $5 < y^+ \leq 30$ . In the buffer layer turbulent eddies are damped quickly. Here, viscous and Reynolds stress contribute to the resulting shear stress. According to Sagaut [104] the mean flow velocity can be described in the buffer layer by:

$$u^+ = A \cdot \ln(y^+) + B. \quad (4.29)$$

The constants can be chosen to  $A = 5.0$  and  $B = -3.05$  [104]. Equation (4.29) is called buffer law in the following.

- The logarithmic layer is defined by  $30 < y^+ \leq y_{max}^+$ . This third layer extends to an upper bound  $y_{max}^+$  which depends mainly on the Reynolds number [94]. Here the total stress  $\tau_{total}$



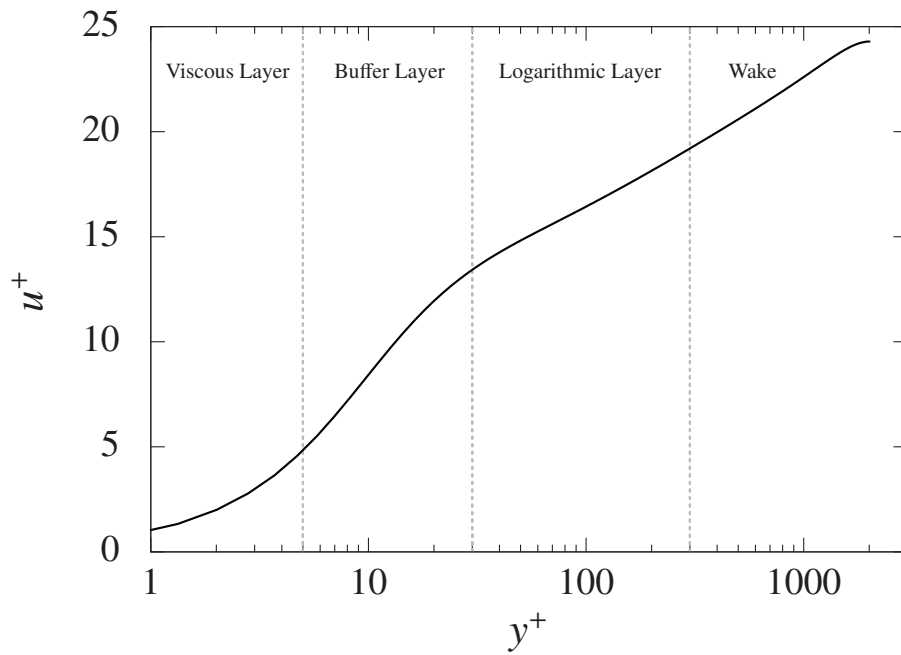


Figure 4.3: Regions of a Turbulent Boundary Layer

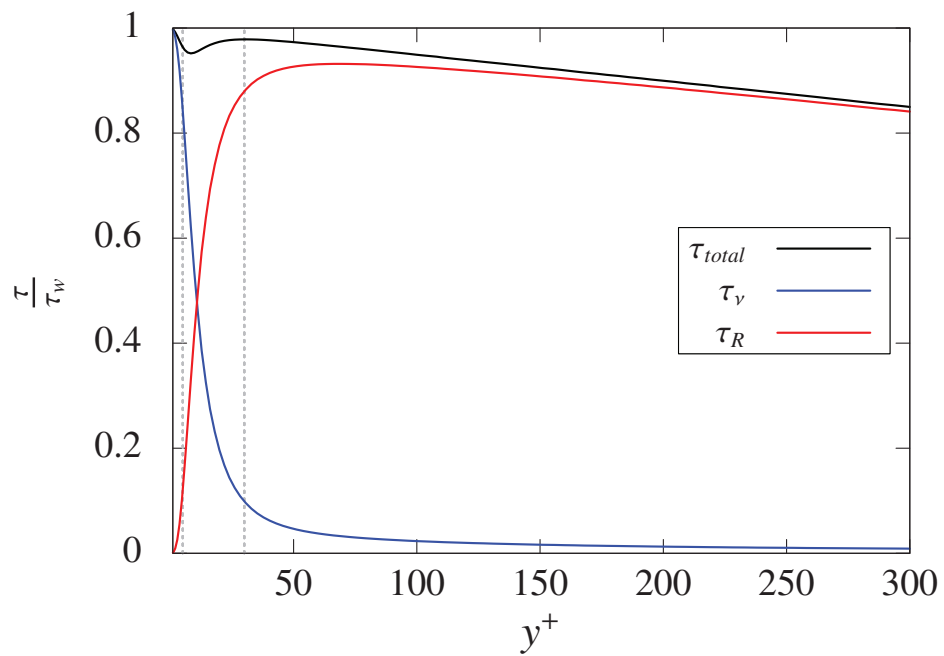


Figure 4.4: Stress Contributions in a Turbulent Boundary Layer

consists basically of the Reynolds stress. At high Reynolds numbers, the length scale of turbulent structures is increasing linearly with the distance to the wall in the logarithmic layer [94]. Neglecting the viscous stress, a correlation for this layer can be achieved from the boundary layer equations, e.g., for channel flows [94]:

$$u^+ = \frac{1}{\kappa} \ln(y^+) + B. \quad (4.30)$$

The constant  $\kappa$  is referred to as von Karman constant, which is usually taken to 0.41, and the other constant  $B$  is typically chosen to  $\sim 5.5$ . Equation (4.30) is commonly named log law.

Besides the inner region, the boundary layer contains the outer region, in which viscosity effects are negligible. At large Reynolds numbers, the inner and outer region can overlap, so parts of the logarithmic layer are element of the inner and outer region. Adjacent to the logarithmic layer, the wake layer follows, which belongs to the outer region. With these details about the boundary layer structure certain requirements for the boundary conditions in LES calculations can be deduced.

Two general concepts for wall boundaries can be subdivided: The wall resolved approach (WR), which performs a LES in the near wall region, and the wall modeled (WM) approach, which applies some kind of model. We will start with the WR approach.

As explained at the beginning of this chapter, LES aims at resolving the large eddies and thus calculates the main part ( $> 80\%$ ) of the turbulent kinetic energy directly. Since the turbulent length scale is rapidly decreasing when approaching the wall, cf. statements above, the grid size must decrease in a similar manner to resolve the large scales of near wall turbulence in the WR approach. Furthermore, the accurate resolution of the velocity gradient near the wall requires that the first grid point is within the viscous layer. From these considerations and numerical practice, it is commonly accepted that the dimensionless wall distance of the first grid point is at  $y^+ \sim 1$  [39, 104]. Conventional CFD methods, which perform LES on non-uniform grids, refine the grid only in wall normal direction and use low aspect ratio grids in the near wall region to save computational requirements. Commonly, the aspect ratios are  $\sim 100$  in streamwise and  $\sim 20$  in spanwise direction. Since the Lattice Boltzmann Method uses equally spaced grids, this treatment is not applicable here and a substantial higher amount of cells is generated. Estimations reveal that the computational demand for resolving the inner region of the boundary layer is of  $O(Re^{1.8})$ , whereas for the outer region it is only  $O(Re^{0.4})$  [39, 104]. These facts make the wall resolved approach computationally expensive and limit its usage nowadays to Reynolds numbers of  $Re \sim 10^5$  [93].

The implementation of the wall resolved approach within the Lattice Boltzmann Method is quite elementary, all Dirichlet conditions introduced in Section 3.5 can be applied. This is due to the fact that the turbulent fluctuations vanish at the wall and that the wall velocity is becoming deterministic. In the WR approach, the subgrid model must be appropriate for describing the

structure of near wall turbulence. For instance, the eddy viscosity should reduce with  $y^{+3}$ , what is realized by using a van-Driest damping function in combination with the Smagorinsky model in SAM-Lattice. Alternatively, this could be done by a dynamic subgrid scale model, as stated above.

The high resolution demand of the inner boundary layer region limits the application of LES for high Reynolds number flows, as explained above. An engineering workaround to bypass this demand is the wall modeling strategy. In WM the first grid point is typically placed in the logarithmic layer and so a huge amount of grid points in the inner region can be saved. Since there is no information calculated in this area, it must be provided by a model now. This model must provide different contributions: Firstly, the distance between the wall and the first node located in the logarithmic layer is larger than the turbulent length scale in this region. So, the influence of the non resolved turbulent structure on the flow must be considered. Furthermore, a critical discussion is needed about the validity and reliability of the LES approach in this case. Secondly, the sharp velocity gradient near the wall can not directly be determined with the first node located that far from the wall. Thus, the wall model must give a relation between local flow quantities and the wall shear stress, i.e., the velocity gradient. An accurate wall shear stress is important to reproduce a correct force balance of the flow.

Let us start, for convenience, with the second point. As it was stated during the definition of the different layers of the inner region, a functional relation between  $u^+$  and  $y^+$  can be achieved for canonical boundary layers. Since both quantities depend on  $\langle \tau_w \rangle$ , this relation provides the desired correlation  $\langle \tau_w \rangle = \mathcal{F}(\langle u \rangle, y, \dots)$ , which is called wall function. The usage of wall functions requires two additional necessities. Due to the definition used here, the wall function is a correlation between the time averaged dimensionless velocity  $\langle u^+ \rangle$  and the time averaged dimensionless wall distance  $\langle y^+ \rangle$ . Both quantities depend on the time averaged wall shear stress. Within the LES context, we need an instantaneous correlation between  $u^+$  and  $y^+$ . As proposed by Werner and Wengle [129] the wall function is assumed to hold instantaneous, so  $\langle u \rangle$  and  $\langle \tau_w \rangle$  are replaced by  $u$  and  $\tau_w$  in the according wall laws. Moreover, the wall function only returns the absolute value of the wall shear stress. Hence, the direction of the stress vector needs to be specified. Here, [129] suggested an in-phase correlation between the present tangential velocity and the wall shear stress:

$$\boldsymbol{\tau}_w = \frac{\mathbf{u}_t}{|\mathbf{u}_t|} \tau_w . \quad (4.31)$$

For complex flows or geometries, respectively, it is not always possible to place the first node during mesh creation a priori in the logarithmic layer, especially when equally spaced grids are used. To ensure a valid behavior of the wall function in cases when the first node is located in the viscous or buffer layer, a piecewise definition of the wall function is commonly

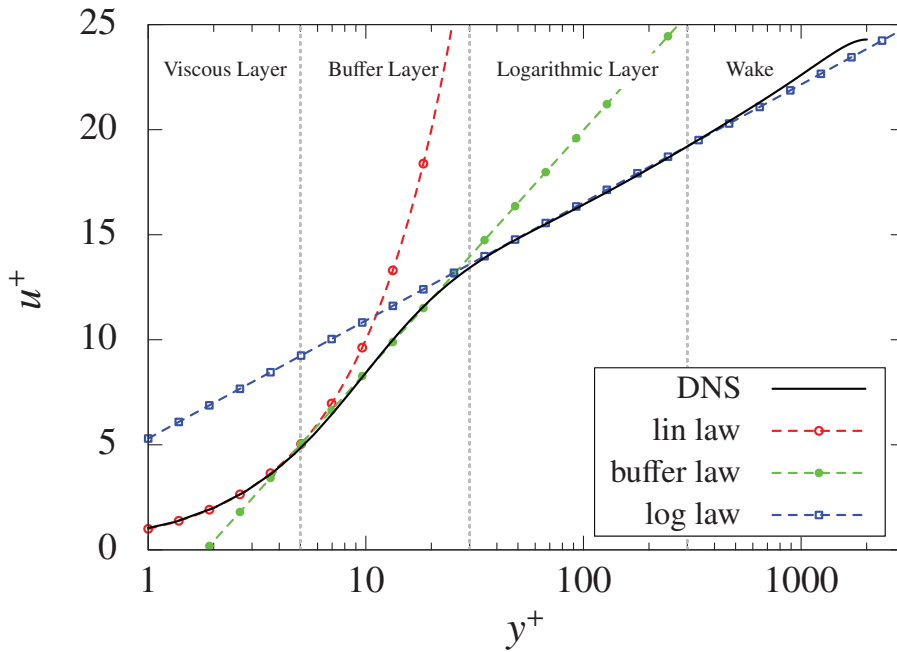


Figure 4.5: Comparison of Log Law Approach

used. From Eqs. (4.28) to (4.30) follows:

$$u^+ = \begin{cases} y^+ & y^+ \leq 5 \\ 5 \ln(y^+) - 3.05 & 5 < y^+ \leq 28.178 \\ \frac{1}{\kappa} \ln(y^+) + 5.5 & y^+ > 28.178. \end{cases} \quad (4.32)$$

Figure 4.5 shows a comparison between the wall function of Eq. (4.32) and DNS data. A good agreement between the approximations in the different layers and the DNS data is observed, whereas the deviation in the wake increases. In contrast to the limiting values in the definition of the layers of the inner region (Fig. 4.3), in Eq. (4.32) the intersection points of the different curves are used as limits to avoid a noncontinuous curve. Nevertheless, the resulting curve contains kinks at the intersection points, due to the different derivatives of the pieces. Instead of the three layer approach of Eq. (4.32), one often finds a two layer approach, which only contains lin law and log law. The switch between these two laws is at the intersection point, cf. Fig. 4.5. The two layer approach is a further simplification, but is disadvantageous if nodes are located in the buffer layer.

The use of Eq. (4.32) entails some difficulties and limitations for arbitrary flows. From the wall function Eq. (4.32), an implicit equation for  $\tau_w$  is recovered, which needs an iterative solution procedure. This can be done, e.g., by Newton's method. An approximation to avoid an iterative procedure is to use the  $y^+$  value from the last time step. But, this can ruin the

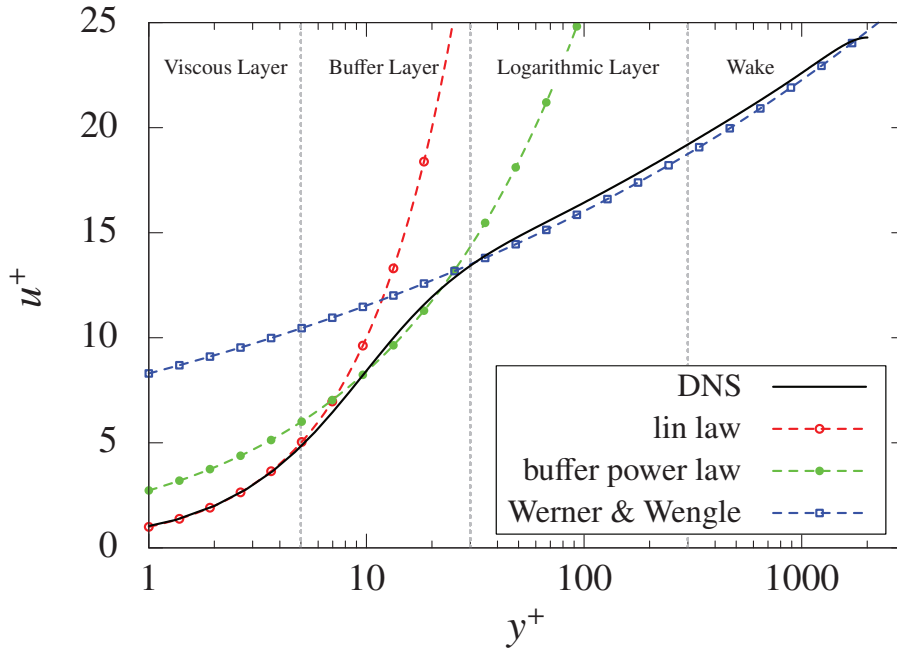


Figure 4.6: Comparison of Power Law Approach

time accuracy. Severe limitations come from the validity range of the wall function. Strictly speaking, the wall function is only valid for canonical turbulent boundary layers, though it is commonly used for all kinds of flows. There are some extension of Eq. (4.32) for adverse pressure gradients and rough surfaces, but they do not compensate all limitations. A more advanced strategy is the solution of a simplified boundary layer equation on a subgrid within the wall and the first node to calculate  $\tau_w$  instead of using a wall function. See, e.g., [93, 104] for details on this technique in the Navier-Stokes context. By following this approach some of the wall function limitations can be eliminated, however the computational and implementation efforts are substantially higher. The implementation of such a boundary layer equation based scheme in SAM-Lattice is recommended for future work.

Werner and Wengle [129] proposed to use a power law based approximation of the velocity profile in the logarithmic layer instead of the log law. Mathematically, the power law is expressed by  $u^+ = Ay^{+B}$ , where  $A$  and  $B$  are constants. The two layer approach of Werner and Wengle [129], which divides the velocity profile in a lin law and a power law, is extended to a three layer wall function here. Thus, the deviation in the buffer layer is reduced:

$$u^+ = \begin{cases} y^+ & y^+ \leq 7.1 \\ 2.73247y^{+0.487215} & 7.1 < y^+ \leq 25.19 \\ 8.3y^{+1/7} & y^+ > 25.19. \end{cases} \quad (4.33)$$

Figure 4.6 depicts the power law wall function of Eq. (4.33) in comparison with DNS data. The deviation of the power law to the DNS is slightly higher than for the log law wall function. But, the power law approach holds the advantage that from Eq. (4.33) an explicit equation for  $\tau_w$  is established. So, there is neither a need for an iterative solver nor for other simplifications. In literature, e.g., [120], the power law reveals excellent results compared with the log law and hence it is used as wall function in SAM-Lattice.

Let us now turn our attention to the modeling of turbulence in WM schemes. As already said, the distance of the first grid node off the wall, which is labeled here with  $y_1$ , is larger than the turbulent length scale, if  $y_1$  is located in the logarithmic layer. This means that all turbulent structures below  $y_1$  in the inner region need to be modeled or the other way around, no structures are resolved. For accurate LES calculations we defined the criterion that at least 80% of the turbulent kinetic energy should be resolved. Accordingly, the requirements for a large eddy simulation are not fulfilled at the first node off the wall, see also [103]. The idea to remove this drawback is using an unsteady RANS approach between the wall and the first grid points, what results in a two layer method. The basic concept in unsteady RANS is only resolving the very largest turbulent motions [38]. More details about RANS in the Navier-Stokes context can be found in, e.g., [34, 38]. As stated by Piomelli [93], the spatial filtering performed in LES will become equivalent to a RANS averaging under certain assumptions. These are the presence of a very large amount of eddies, located between the wall and the first node, and a time step which is much larger than the turbulent time scale near the wall. Concerning the mathematical structure of the governing equations, there are no differences between the filtered LBE and the unsteady RANS averaged LBE, if both are based on the eddy viscosity assumption. The same holds for the classical Navier-Stokes framework, as shown by [38]. The first RANS-LBM model was proposed by Teixeira [119]. Although, there are no differences in the structure of the equations, we should keep the differences in the physical modeling in mind.

The unsteady RANS approach requires a different consideration of the turbulence. Again, the eddy viscosity assumption is used, but now for modeling the Reynolds stress tensor instead of the subgrid stresses. For the near wall layer, the eddy viscosity will be based on a mixing length model, since this model is computational less demanding and easy to implement. In principle other statistical models can be applied. The definition of mixing length model is [131]:

$$\nu_t = l_m^2 \left| \frac{\partial u}{\partial y} \right|. \quad (4.34)$$

There are different definitions of the mixing length  $l_m$  proposed in literature. Balaras et al. [8] used the following equation:

$$\nu_t = (\kappa y)^2 \left| \frac{\partial u}{\partial y} \right|. \quad (4.35)$$

In Eq. (4.35)  $\kappa$  again represents the von Karman constant and  $y$  is the distance to the wall. The derivative of the velocity cannot be calculated accurately in the numerical model, due to the coarse grid resolution and the sharp gradients. Instead, the derivative is calculated from the

wall function (Eq. (4.33)). By inserting the derivative of the log law, Eq. (4.35) reduces to:

$$\nu_t = \kappa y u_\tau . \quad (4.36)$$

For the power law approach  $\nu_t$  results in:

$$\nu_t = \kappa^2 AB \frac{u_\tau^{B+1}}{y^B} . \quad (4.37)$$

The coefficients  $A$  and  $B$  can be found in Eq. (4.33). Both formulations of  $\nu_t$  can be combined with the van-Driest damping ( $\nu_t^{y^D} = \nu_t (1 - \exp(-y^+/A^+))^2$ ) to reduce the eddy viscosity in the buffer layer. In the viscous layer  $\nu_t$  is commonly neglected [103].

In the last part of this subsection the application of the wall model to the Lattice Boltzmann Method is explained. Wall functions have been used in the Lattice Boltzmann context in combination with RANS models [36, 119]. To the best of the author's knowledge, this wall modeled two layer LES approach is an innovation. The implementation of the two layer approach in LBM is readily described by means of Figs. 3.7 and 3.8, respectively. First of all, the velocity is interpolated at the point  $\mathbf{x}_{I1}$  and decomposed into its normal and tangential  $\mathbf{u}_t(x_{I1})$  part. With the tangential velocity and the normal distance  $h$  to the wall, the absolute value of the wall shear stress is obtained from the following relation, which is derived from Eq. (4.33):

$$\tau_w = \begin{cases} \sqrt{\frac{u_t \nu \rho}{h}} & y^+ \leq 7.1 \\ \rho \left[ \frac{u_t}{2.73247(h/\nu)^{0.487215}} \right]^{\frac{1}{1.487215}} & 7.1 < y^+ \leq 25.19 \\ \rho \left[ \frac{u_t}{8.3(h/\nu)^{1/7}} \right]^{\frac{1}{8/7}} & y^+ > 25.19 . \end{cases} \quad (4.38)$$

Knowing the absolute value of  $\tau_w$  according to the wall function, one of the proposed shear boundary conditions in Subsection 3.5.7 can be used. The implementation of the shear boundary conditions already uses the in-phase correlation for the shear stress and the tangential velocity (Eq. (4.31)). Due to numerical stability advantages the shear anti bounce back is preferred in the following.

The second step is the specification of the eddy viscosity according to the mixing length model, instead of the Smagorinsky model. So, all fluid nodes adjacent to the wall have to calculate  $\nu_t$  according to Eq. (4.37).

### 4.3.2 Inlet Boundary Conditions

Specification of inlet boundary conditions in LES brings up some open questions. In the case of laminar flow at the inlet, the problem can be solved easily. We can use the Dirichlet boundary conditions introduced in Section 3.5 to specify velocity or pressure conditions at the boundary under these circumstances. In the case of turbulent flow at the inlet boundary, the flow cannot be described deterministically. A solution is sought for generating appropriate turbulent inflow

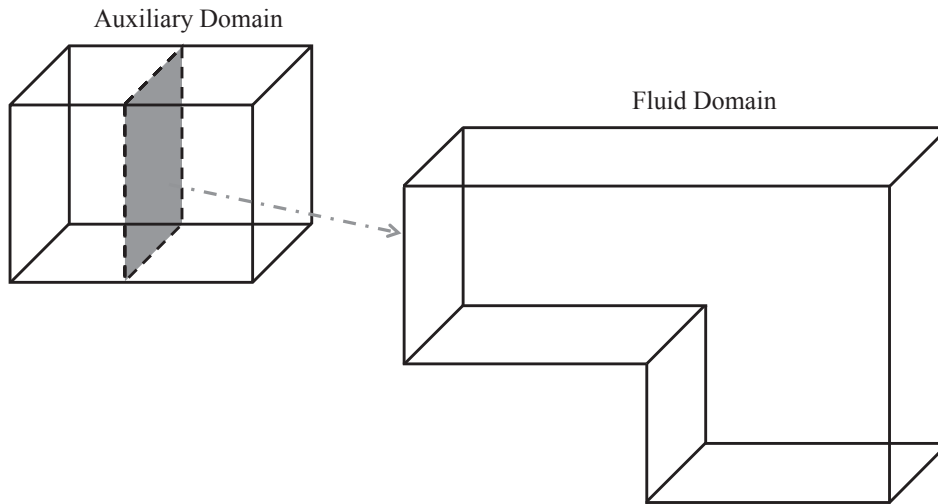


Figure 4.7: Schematic Representation of a Precursor Simulation

conditions, since the inflow conditions have a crucial influence on the flow. The easiest and most impractical way to solve this problem is to extend the fluid domain at the inlet by a transition region. At the extended inlet a mean velocity profile is specified afterwards and the flow can emerge to full turbulence before reaching the previous inlet position. This approach is nearly unfeasible, since the transition region can be quite long and causes high computational costs. Other trivial possibilities are to use velocity fields from measurements or DNS calculation, but for general flow problems these data are not available. In this subsection two possible methods will be introduced which are used in SAM-Lattice to provide appropriate inlet conditions.

The first approach is called *precursor simulation*. The basic idea is similar to the extension of the domain at the inlet. But instead of a domain extension, a second auxiliary domain is used, which is rendered periodically in streamwise direction. By means of the periodicity condition, the effective length of the domain can be reduced. The auxiliary domain has the same cross section as the inlet boundary and should have a grid matching to the inlet plane to avoid interpolation. After the flow field in the auxiliary domain has reached the desired state, the calculation of the actual domain can start. From a cross section of the auxiliary domain, the velocity field is exported every time step and used as inlet boundary condition for the fluid domain, as exemplary depicted by Fig. 4.7. The main disadvantage of this approach is the computational demand for the auxiliary simulation, which is generally lower than extending the fluid domain, but can still be quite high. Furthermore, the velocity information at the auxiliary cross section needs to be stored for an adequate amount of time, which should be in the order of the integral timescale of the flow [104]. After one integral timescale has past, the velocity from the beginning can be reused. Two last constraints to mention here are that the flow in the auxiliary domain must of



course be applicable to a periodic treatment and that the streamwise domain size must be large enough to admit realistic turbulent structures.

To enable simulations of complex flows of engineering interest, a more sophisticated scheme with affordable computational demand is needed. The schemes which are most promising for such cases are based on synthetic turbulence, see [41] for an overview. Up to now, the topic of synthetic turbulence boundary conditions has found only little attention in the LBM context. As far as the author is aware of, only [97] recently used a member of this class, developed in [121] for modeling of turbulent structures at the inlet in LBM. We propose to use a different, more recent scheme, the synthetic eddy method (SEM), within the Lattice Boltzmann Method. SEM was developed by Jarrin et al. [61] in the Navier-Stokes context. Advantages of SEM are, among others, its low computational costs and the ability to work on arbitrary shaped inlets and meshes, what suits ideal to the approach followed by SAM-Lattice. In the following, the principles of SEM and its implementation are introduced.

The basic concept of SEM relies on constructing coherent structures, i.e., synthetic eddies in the physical domain. From the superposition of these eddies, the turbulent fluctuations are obtained. This reflects the classical interpretation of turbulence as a superposition of coherent structures. By this simple concept, SEM differs from most representatives of synthetic turbulence schemes, which rely on adding perturbations to certain wave numbers in spectral space.

The first step in the implementation of SEM is the creation of the box of eddies. In this ashlar-formed box the synthetic eddies are constructed and their dynamics are modeled. The position and the dimensions of this box are determined by the set of grid nodes of the inlet boundary  $B_I \subset \mathbb{R}^3$ , see Fig. 4.8, and the desired length scales  $l_{ij}(\mathbf{x})$  of the turbulent structures according to Eq. (4.39). Anisotropic turbulence is created by choosing different length scales  $l_{ij}(\mathbf{x})$  for each velocity component  $i$  and axes directions  $j$  and isotropic turbulence by setting  $l_{ij}(\mathbf{x}) = l = \text{const.}$

$$x_{j,\min} = \min_{\mathbf{x} \in B_I, i \in \{x,y,z\}} (x_j - l_{ij}(\mathbf{x})) \quad \text{and} \quad x_{j,\max} = \max_{\mathbf{x} \in B_I, i \in \{x,y,z\}} (x_j + l_{ij}(\mathbf{x})) \quad (4.39)$$

In the second step the box is filled with eddies. For this procedure, the number of synthetic eddies  $N$  in the box must be defined. Since the turbulent kinetic energy in the box must be a constant, which is independent of the number of eddies, the energy per eddy is decreasing for increasing  $N$  and vice versa. In [60] the influence of  $N$  on the resulting turbulence statistics was analyzed. The result of this study is that the LES solution becomes independent of the number of eddies for  $N$  greater than a certain value, which can be estimated from the ratio of the eddy box volume  $V_B$  and the turbulent length scales  $l_{ij}(\mathbf{x})$ :

$$N = \max_{i \in \{x,y,z\}} \frac{V_B}{l_{ix}(\mathbf{x})l_{iy}(\mathbf{x})l_{iz}(\mathbf{x})}. \quad (4.40)$$

To minimize the computational costs, Eq. (4.40) is used in SAM-Lattice for the definition of  $N$ . The initial positions  $\mathbf{x}_k$  of the eddies in the box are set independently by a random operation.

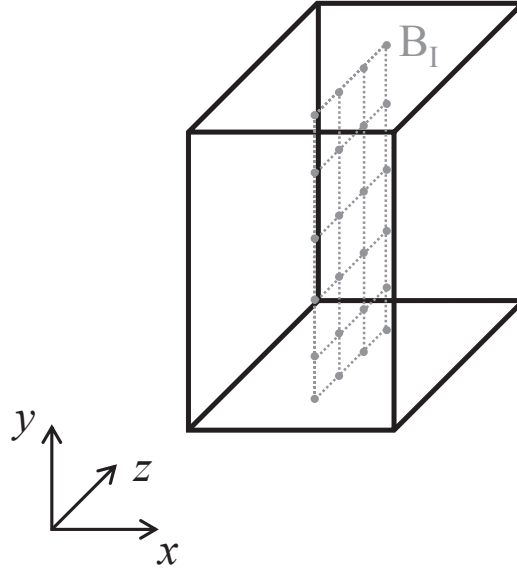


Figure 4.8: Box of Eddies

Additionally, every eddy holds a three component vector  $\boldsymbol{\varepsilon}_k$ , which contains the orientation of the eddy in the three space directions. The components of  $\boldsymbol{\varepsilon}_k$  are set independently by random to  $-1$  or  $1$ . With this step the initialization procedure is completed and the calculation of the turbulent fluctuations can start.

Step number three is the calculation of the turbulent fluctuations at an inlet node  $\mathbf{x}_I$ . The turbulent fluctuations result from a superposition of the synthetic eddies in the box [61]:

$$u'_i = \frac{1}{\sqrt{N}} \sum_{k=1}^N c_{k,i} f_{l_{ij}}(\mathbf{x}_I - \mathbf{x}_k). \quad (4.41)$$

The quantity  $c_{k,i}$  represents the intensity of  $k$ -th eddy in spatial direction  $i$ :

$$c_{k,i} = a_{ij} \varepsilon_{k,j}. \quad (4.42)$$

$\mathbf{A} = a_{ij}$  is the Cholesky decomposition of the Reynolds stress tensor  $\mathbf{R}$  which should be met by the synthetic turbulence [61].

$$A = \begin{pmatrix} \sqrt{u'u'} & 0 & 0 \\ u'v'/a_{11} & \sqrt{v'v' - a_{21}^2} & 0 \\ u'w'/a_{11} & (v'w' - a_{21}a_{31})/a_{11} & \sqrt{w'w' - a_{31}^2 - a_{32}^2} \end{pmatrix} \quad (4.43)$$

The velocity fluctuation created by each eddy in Eq. (4.41) is controlled by the shape function  $f_{l_{ij}}(\mathbf{x}_I - \mathbf{x}_k)$  of the  $k$ -th eddy at position  $\mathbf{x}_k$  [60]:

$$f_{l_{ij}}(\mathbf{x}_I - \mathbf{x}_k) = \sqrt{V_B} \frac{1}{\sqrt{l_{ix}}} f\left(\frac{x_I - x_k}{l_{ix}}\right) \frac{1}{\sqrt{l_{iy}}} f\left(\frac{y_I - y_k}{l_{iy}}\right) \frac{1}{\sqrt{l_{iz}}} f\left(\frac{z_I - z_k}{l_{iz}}\right) \quad (4.44)$$

$$f(x) = \begin{cases} \sqrt{\frac{3}{2}} (1 - |x|) & |x| \leq 1 \\ 0 & |x| > 1. \end{cases} \quad (4.45)$$

Instead of the tent function Eq. (4.45), other shape functions can be used, see [60] for examples. But, all shape functions must fulfill the normalization constraint to conserve the turbulent kinetic energy in the box:

$$\int_{-1}^1 f^2(x) dx = 1. \quad (4.46)$$

The above definitions ensure the desired turbulence statistics  $\langle u'_i \rangle = 0$  and  $u'_i u'_j = R_{ij}$ . A detailed proof of these single point correlations and an investigation of further correlations can be found in [60]. The calculus of this third step enables the determination of the velocity fluctuations at all inlet nodes.

In the fourth and last step of SEM the dynamics of the eddies are processed. In SEM it is assumed that the eddies convect with a characteristic velocity through the box [61] according to Taylor's hypothesis. This characteristic velocity  $\mathbf{u}_c$  is defined as the average velocity of all inlet nodes. Within one time step the center of an eddy moves in the box from  $\mathbf{x}_k(t)$  to  $\mathbf{x}_k(t + \Delta t)$ .

$$\mathbf{x}_k(t + \Delta t) = \mathbf{x}_k(t) + \mathbf{u}_c \Delta t \quad (4.47)$$

If the center of an eddy is leaving the box during this convection process, it is recreated on the opposite surface ( $F$ ) of the box through which it left. This recreation process contains some specifics. The position of recreated eddy on the face  $F$  is set randomly and also the orientation vector  $\boldsymbol{\varepsilon}_k$  is again set by random.

The creation of turbulent structures by SEM at the inlet of a LES domain is achieved by performing the steps 3 and 4 of the procedure every time step of the calculation. Steps 1 and 2 of the procedure need only be performed once during the initialization. Inclusion of SEM into LBM for velocity inlet conditions is straight forward. The velocity fluctuations generated by SEM are superimposed to the prescribed velocity at the inlet node to a resulting velocity at this node. With this resulting velocity, one of the Dirichlet boundary conditions for velocity, introduced in Section 3.5, can be used.

One problem that remains is the requirement of SEM concerning the turbulence characteristics. In most cases one will neither know the entire Reynolds stress tensor to calculate the Cholesky decomposition (Eq. (4.43)) nor the different turbulent length scales  $l_{ij}(\mathbf{x})$  at the domain inlet. This problem can be reduced, under the assumption of isotropic turbulence, to the commonly known problem of RANS calculations, which need specification of two turbulence

properties, e.g., turbulence intensity and length scale. In the case of isotropic turbulence, we have  $u' = v' = w'$  and the Cholesky decomposition reduces to  $a_{11} = a_{21} = a_{31} = |u'|$  and all other  $a_{ij} = 0$ . The remaining quantity  $u'$  can easily be calculated from the turbulence intensity  $I = u' / |\bar{\mathbf{u}}|$ , for which different empirical correlations are available in literature. As said before, also the turbulent length scales are equal in the isotropic case  $l_{ij}(x) = \text{const} = l$ . For the estimation of  $l$  empirical correlations can be found in literature, too.

## 4.4 Grid Refinement for Large Eddy Simulation

The grid refinement as introduced in Section 3.7 needs some special considerations in the case of large eddy simulation. As explained at the beginning of this chapter, the cutoff length of the filtering operation is commonly associated with the grid spacing  $\Delta x$ . A change of the grid spacing thus results in different cutoff lengths and different resolved and subgrid scales on the various grid levels. These facts have to be taken into account for the interface treatment. More precisely, we only need to focus on the scaling operations of interface nodes. The interpolation procedure of the hanging nodes can be ignored since this operation uses only values from the same fine grid level.

Let us start with the scaling process from fine to coarse. Here the distribution functions from the fine grid level are used to calculate the distribution functions on the coarse level. Since we have resolved scales in the fine level quantities, which cannot be represented on the coarse level, the fine level quantities need an additional treatment before scaling. This procedure is known from the Navier-Stokes context as restriction [104]. It is performed by an additional filtering operation of the fine level quantities, which restricts them to the coarse level. Numerically, this filtering is realized in SAM-Lattice as averaging  $f_i^f$  over all lattice neighbors of the fine partner node before scaling (see [71, 99] for details of the discrete filtering operation):

$$f_i^f(\mathbf{x}, t) = \frac{1}{q} \sum_{j=0}^{q-1} f_i(\mathbf{x} + \boldsymbol{\xi}_j \Delta t, t). \quad (4.48)$$

[128] proposed a different form of the discrete filtering, in which the distributions are weighted according to their lattice direction. The difference in the resolved scales is directly accompanied by a difference in the subgrid scales on the fine and coarse level, respectively. For this reason the eddy viscosity, which models these scales, cannot simply be copied from the fine level to the coarse level. From the Smagorinsky model (Eq. (4.12)), we see that the eddy viscosity depends on  $\Delta x^2$ . Assuming the change of the norm of the resolved strain rate tensor  $|\bar{\mathbf{S}}|$  on the different levels is small and negligible, the eddy viscosity on the coarse level is four times the eddy viscosity on the fine level. That results for the applied refinement ratio of two in:

$$\nu_t^c = 4\nu_t^f. \quad (4.49)$$

Recently [128] proposed the use of the following relation between the eddy viscosity on the fine and coarse level, which comes from a dimensional analysis (see also [104]):

$$\nu_t^c = (2)^{4/3} \nu_t^f. \quad (4.50)$$

Equation (4.50) has not yet been tested by the author, but comparison with Eq. (4.49) shows a difference of approximately 37 % between both formulations.

With the filtered distributions at the fine partner node (Eq. (4.48)) and the rescaled subgrid viscosity (Eq. (4.49)), the scaling from fine to coarse can be performed, as described in Section 3.7.

For the scaling from coarse to fine we have the opposite case. The distributions available on the coarse level, which are used for scaling to the fine level, do not include length scales smaller than  $\Delta x_c$ . The task here is to create scales in the length scale between  $\Delta x_c$  and  $\Delta x_f$  for the quantities on the fine level. This is known as enrichment in literature [104]. From a mathematical point of view this can be interpreted as a deconvolution procedure and the known deconvolution schemes can be applied. Another way to create these small scales, i.e., high frequency parts, is the use of synthetic turbulence methods as introduced in the subsection dealing with inlet boundary conditions. SEM could be used for this problem, even though an additional filtering operation of the synthetic fluctuations is necessary to obtain only scales between  $\Delta x_c$  and  $\Delta x_f$ . A more convenient way is to extrapolate the fluctuations in the desired scale range from the neighbors of the fine interface nodes and to use these at the interface; see [99] for details of the extrapolation procedure.

Currently, no enrichment procedure is used in SAM-Lattice, since the enrichment procedure is a demanding process and often leads to numerical instabilities as experienced by the author and also stated in [128]. Nevertheless, the eddy viscosity of the coarse partner nodes is rescaled for the fine interface nodes. This is done from Eq. (4.49), thus the eddy viscosity at the fine interface nodes corresponds to one quarter of the eddy viscosity at the coarse partner nodes.

## 4.5 Practical Aspects for Large Eddy Simulation

Practical application and result evaluation of large eddy simulation hold some difficulties which will be shortly addressed. As we already said a couple of times, a sound LES should resolve at least 80% of the turbulent kinetic energy  $k$  [94].

In engineering applications, numerical simulations are often used as a prediction tool in the design process. Thus, the flow field and the total amount of  $k$  are unknown, what complicates the assessment of uncertainty in LES. In the RANS framework commonly grid studies are performed for quality assessment. For LES, there is theoretically no grid independency, since the filter width is associated to the grid size. Hence, a grid independent LES is essentially a DNS, i.e., all scales are resolved [16]. In literature, different grid resolution indices have been proposed to enable quality assessment if no validation data is available, see [40] for an overview.

One class of these indices are the single-grid estimators, which require only the solution of one LES calculation. These estimators can be applied after a first LES run and used as a decision-maker, if further grid refinement is necessary. The simplest single-grid estimator is the ratio of eddy viscosity to the molecular viscosity  $\nu_t/\nu$ , which will be used in this thesis. The physical meaning behind this estimator is that  $\nu_t$  represents the subgrid scales and the modeled part of  $k$ , respectively. A drawback of this easy to use estimator is the dependence of the maximum acceptable value on the Reynolds number [40]. Depending on the reference consulted, this ratio should be less than 25 [16]. Another often used single-grid estimator relies on the power spectrum of  $k$  in the frequency or wave number domain, respectively. From the slope of this quantity we can see if the inertial range of the turbulent spectrum, characterized by the  $-5/3$  slope, is recovered. More advanced grid estimators relying on a single or multiple solutions can be found in [16, 40].

Another important point to mention here is the surprising fact that the resolved turbulent kinetic energy in LES can be larger than  $k$  resulting from DNS or experiments. This numerical artifact is mainly justified in literature with poor resolution of the LES, see [16] for more detailed attempts of explanation.

---

### Implementation Aspects of the CFD Package SAM-Lattice

---

The implementation of the previously described Lattice Boltzmann Method is done by an object oriented approach in C++. The object oriented programming paradigm has been chosen due to its advantages, which are the possibility of real-world modeling and the reusability of code, among others. C++ is used since it is object oriented, includes template capabilities, is supported on many platforms, and provides many libraries. In the course of this thesis, a modular structured software package called SAM-Lattice has been developed by the author and his colleague Dipl.-Ing. Daniel Conrad.

#### 5.1 Conceptual Aspects

The keynote in the design of the software package SAM-Lattice is to create a CFD tool based on the Lattice Boltzmann Method which is applicable to complex, real-world problems of engineering interest in 3D. This diversified and generalized area of application demands an elaborated and sound design of the software. Further development objectives are user-friendliness, high efficiency and automation. To achieve these goals, we decided to use a modular structure of the package, consisting out of five components, see. Fig. 5.1. The module SamLattice is a graphical user interface (GUI), developed with QT, which allows the utilization and control of the remaining package-modules and also includes features for on-line result monitoring. Sam-Lattice is devoted to user-friendliness, since the remaining four components can be used and controlled without the GUI by scripts, e.g., for batch computing. As common in CFD applications the workflow of a simulation in SAM-Lattice is subdivided into the consecutive steps of preprocessing, solving, and postprocessing. For preprocessing two modules are included in SAM-Lattice. The SamGenerator is a highly automated lattice generator for the discretization

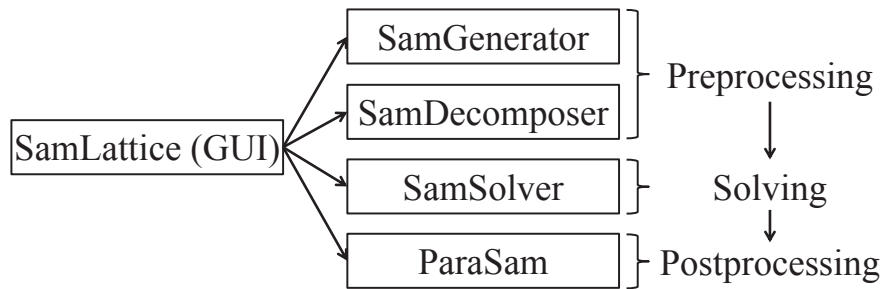


Figure 5.1: Modules of SAM-Lattice

of the domain. For distributed memory parallelization, a domain decomposition is necessary, which is provided by the SamDecomposer. SamDecomposer makes use of the graph partitioning library METIS for an optimal decomposition of the discretized domain. The actual implementation of the Lattice Boltzmann Method as a numerical scheme for fluid dynamical problems is included in the solver SamSolver. After the solution process is completed, result evaluation can be performed with the postprocessor ParaSam, which is our customized version of the open-source visualization tool ParaView, which is based on the visualization tool kit (vtk) and developed by Kitware. ParaSam contains some additional, in-house developments for input/output operations and analysis features.

Every module of SAM-Lattice, except the GUI, requires particular input data and produces specific output data, which is the basis for the next step of the workflow. Through this file management, it is always possible to go one step back in the workflow and modify certain configuration parameters without the need to restart completely from scratch. In the remaining part of this chapter some implementation aspects of the two core modules of the package, SamGenerator and SamSolver, will be presented. Precise details of the implementations can be found in the detailed programmer's guides of the corresponding modules.

## 5.2 Preprocessor - SamGenerator

As already said, SamGenerator is the discretization module of SAM-Lattice. It calculates the computational grid for the numerical solution based on the Lattice Boltzmann Method. The discretization relies on the uses of regular grids, which can be refined in certain areas, as introduced in Section 3.7. Figure 5.2 depicts the basic operations in terms of subroutines, which are performed during a SamGenerator run. To get a basic understanding how SamGenerator is working, we will follow the flowchart.



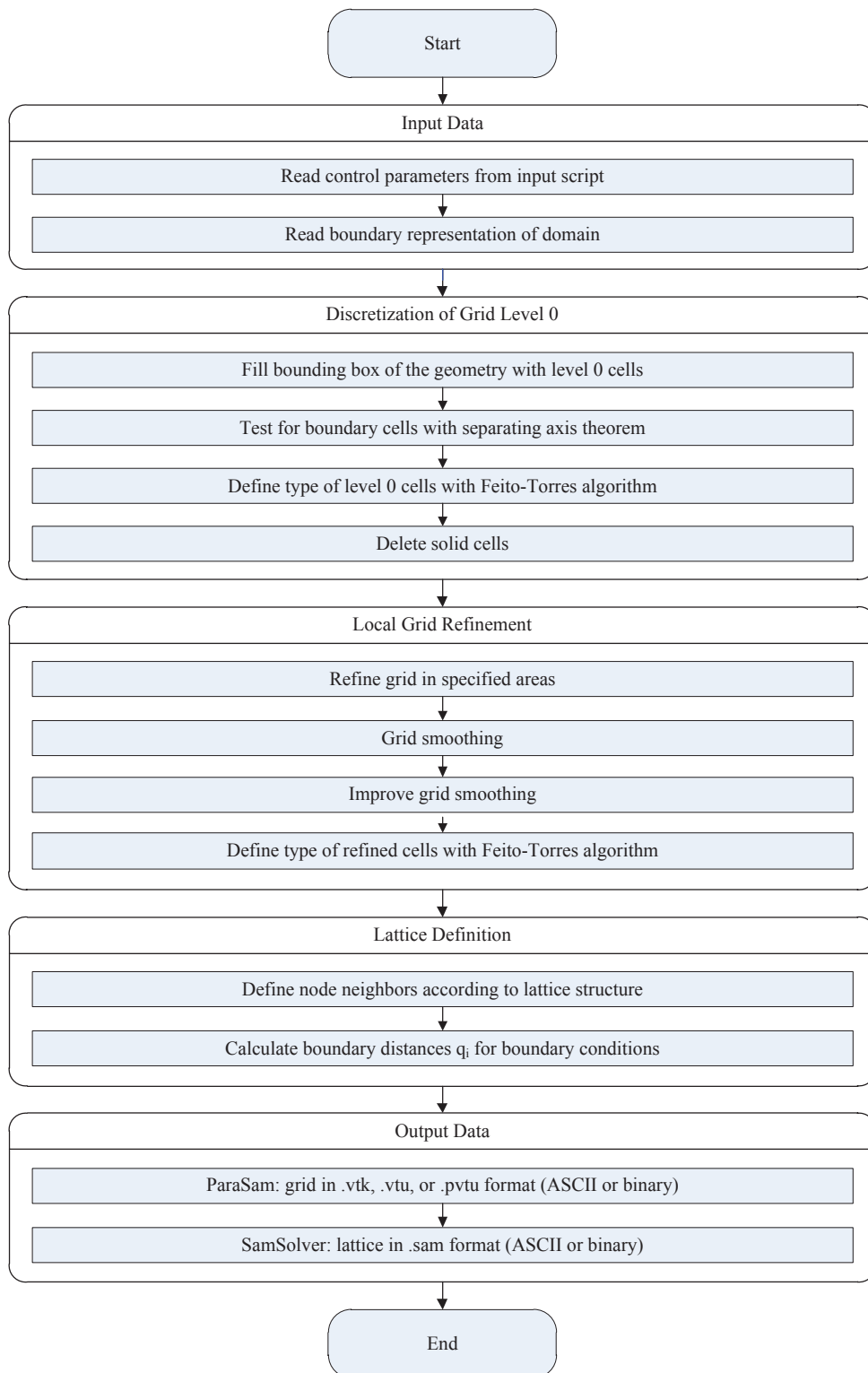


Figure 5.2: Flowchart of SamGenerator

## 5.2.1 Input Data

The first conceptual block of routines is reading of input data, what consists of control or mesh parameters, respectively, and the domain geometry. SamGenerator requires the fluid domain in 3D as geometry input. Geometrical modeling is not part of the capabilities of SAM-Lattice, but can be done by available CAD systems. The geometry input for SamGenerator must rely on boundary representation (BREP), i.e., the geometry is completely characterized by its surrounding boundaries. The boundaries itself must be represented by triangulated surface meshes. Currently, two standard file formats are supported for geometry representation: msh [7] and stl [14] format. In general the msh format should be preferred since it includes by default information like boundary names and their physical types, which are mandatory for the grid generation. If the stl format is used, it must be extended by these information.

## 5.2.2 Discretization of Grid Level 0

After reading input data, the discretization process starts at level 0. The bounding box of the geometry input is calculated and filled with cubes, which are sized according to the user specification. To represent the geometry by this basic grid, we need to figure out, which cells and nodes are in-/outside of the fluid domain and which cells are intersected by the fluid domain boundaries. Such problems typically occur in computer graphics and very efficient algorithms have been developed for their solution. We use a smart combination of different algorithms within SamGenerator. First of all an intersection test between the cells and the triangles of the BREP is performed. This test can be efficiently performed by the separating axis theorem (SAT), which states that two convex polyhedra are disjoint if an axis can be found, on which the projections of these two objects do not overlap [2]. See also [22] for more details on the implementation in SamGenerator. As a result of SAT, we know which cells are intersecting the boundaries and call them boundary cells. But, we do not know which nodes of the intersection cells are inside or outside of the domain. By definition, a node on the boundary itself is outside. Furthermore, we do not know if the remaining, non-intersection cells and their nodes are inside or outside. This problem is known in computational geometry as point inclusion test in polyhedra. A fast, efficient and robust scheme for inclusion tests in general polyhedra is the Feito-Torres method [33], which will be shortly introduced. In the Feito-Torres method, for each triangle of the boundary mesh a tetrahedron is built with the origin as fourth point. If a triangle contains already the origin as a point, an arbitrary translational transformation of the triangle is performed. To determine if a point  $P$  is now inside a polyhedron, the signs of the tetrahedron volumes, which are build from the boundary mesh and containing  $P$ , are summed.  $P$  is inside of the polyhedra, if and only if the resulting sum is  $> 0$ . It can be shown that this algorithm fulfills the Jordan curve theorem [33] and is thus suitable for point inclusion tests. By some preprocessing operations the Feito-Torres method can be dramatically optimized [90].

To further decrease the computational costs, the Feito-Torres method is used in combination with a flood fill algorithm. The idea behind this combination is that it is not necessary to

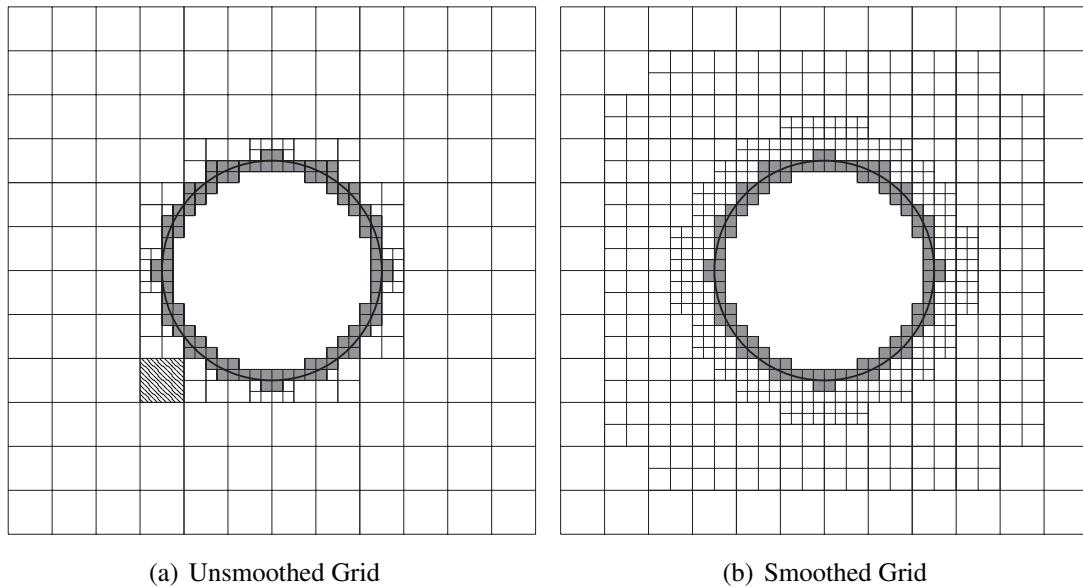


Figure 5.3: Grid Refinement and Smoothing

test every point with the inclusion test, since we know the intersection cells. In the following, nodes inside the domain will be called fluid and outside nodes are named solid. This means we perform the inclusion test for one grid node of a non-intersection cell and determine if it is inside or outside. Since it is a non-intersection cell, all other nodes of this cell must have the same type (fluid/solid), i.e., the whole cell is set to fluid or solid. Even more, we can march in all directions of space, starting from this cell, and set the according neighbor cells to the same type, until we encounter a boundary cell. This makes a further inclusion test necessary, since the node type can change now. By this methodology the necessary inclusion tests are decreased to a minimum. The node types of the boundary cells are determined by the connectivity to the neighboring cells or, if this is not possible by further inclusion tests.

The last step of the grid generation on level 0 is the deletion of solid cells and nodes, which do not participate in the simulation. With this sparse representation of the grid the memory demand is reduced, what reveals especially for porous media or domains including large holes or obstacles. On the other hand, this step increases the complexity dramatically. The data structure must allow the removal of unused cells and their according nodes. This calls for an unstructured grid layout, which is realized in SamGenerator with help of the map container of the C++ standard template library.

### 5.2.3 Local Grid Refinement

As already stated in Section 3.7 the grid refinement in SAM-Lattice relies on an octree structure, so for refinement a parent cell is divided into eight regular child cells. There are different ways

to specify the regions, which are to refine. One way is boundary related refinement. Here, the user can define at which boundary the grid should be refined and also the refinement depth, i.e., how many levels are used, can be assigned. The other possibility is region related refinement. In this way, the user defines a region by a rectangular box or an arbitrarily shaped geometry, via an BREP of this region. Inside of the box or the geometry, represented by the BREP, the grid is refined up to the desired level. Since we are using a tree data structure, our refinement technique can be conceptually classified as a top down approach, because we recursively subdivide specific cells starting from the root (level 0) of the tree. The result of a boundary refinement process is exemplary shown in 2D for a sphere in Fig. 5.3(a).

The next routine in the conceptual block of grid refinement in Fig. 5.2 deals with the smoothing of the grid, i.e., this algorithm ensures that the level difference of neighboring cells is not greater than one. A violation of this requirement is highlighted in Fig. 5.3(a). Furthermore, the subroutine warrants that the minimum width of cells per grid level and near boundaries are realized. Both quantities are user inputs. By default, the minimum number of cells in every direction is three fine cells. For boundary refinement, the default minimum number of refined coarse cells next to the wall is three. These default values are necessary to prevent an overlapping of the interface regions of different levels. Arithmetically, the smoothing operation is realized by parsing the octree structure and checking if the mandatory child cells (leaves) are present. Missing child cells are created by calling the refinement routine for the according coarse cells. The final grid for the example of Fig. 5.3(a) after the smoothing operation is depicted in Fig. 5.3(b).

With the end of the grid smoothing, the mesh is according to the requirements of the numerical method and ready to use. Nevertheless, from a numerical point of view it is advisable to avoid unnecessary interpolations, which can derogate the accuracy and numerical stability. For these reasons we want to minimize the surface of the interface by further refinement. This is done by an additional, iterative smoothing process based on rules defined in [25]. The maximum number of iterations of the improve smoothing routine can also be set by the user.

Finally, we must determine the type of the refined cells and nodes. This is again done with the Feito-Torres' method. To reduce the calls of this routine and save thereby computational costs, we use again the mesh connectivity. The refinement of a fluid cell, for instance, must result in fluid child cells containing only fluid nodes. In boundary cells, we test the child cells by the SAT algorithm first. Doing this, we know which child cells are boundary cells itself and which are not. The type of non-boundary cells and their nodes can be determined from the node type of the common node with the parent cell. In this fashion the necessary point inclusion tests for the refinement process are considerably reduced.

## 5.2.4 Lattice Definition

So far, we calculated a regular grid with local refinement, which is not specific for the Lattice Boltzmann Method. The intrinsic information needed for the Lattice Boltzmann algorithm is obtained in next conceptual block of Fig. 5.2. As we rely on an unstructured grid represen-

tation, we must find the neighbors of every node in the particular lattice, which is D3Q19 in SAM-Lattice. During this operation we also associate the boundary nodes to their boundary conditions, as specified in the geometry definition.

The second Lattice Boltzmann specific algorithm is the calculation of the boundary distances  $q_i$  according to Eq. (3.48), which are used for the second order accurate boundary treatment.  $q_i$  results from the intersection of the lattice links with the triangles of the BREP and can thus be calculated from geometrical considerations. Details of the implementation in SamGenerator are provided by Conrad [22]. With this step the lattice calculation is completed.

### 5.2.5 Output Data

Due to the modular design of SAM-Lattice, the grid information is not internally handed to the solver and must be written to a file. Of course, this produces some input/output (I/O) overhead, but enables multiple calculations with the same mesh without the need of multiple mesh calculations. The output block contains mainly two routines. One routine exports the grid in a vtk-compatible format for ParaSam. At the moment, the legacy vtk-format or the xml-based vtu and pvtu formats are supported for ASCII as well as for binary file formats. In the vtk-compatible formats no overlapping of refined cells is included. This means, if a cell is refined, only the leaves, i.e., the most refined cells are exported. The rest of the octree is removed.

The second routine provides the export capabilities to SamSolver. For data transfer to SamSolver a new file format (.sam) has been designed. This includes besides the standard mesh structure the lattice specific information of the D3Q19 neighbors, boundary distances  $q_i$ , and the definition of the level interfaces. The sam-file is supported in ASCII and binary format. As for the vtk-compatible format, only the most refined cells are exported, except for the interface regions, where an overlapping of one coarse cell width is needed for algorithmic reasons, see Section 3.7. Of course, the sam format can also be handled by ParaSam for visualization purposes. After the file export, the lattice generation is completed and the run of SamGenerator ends.

There is another export routine implemented in SamGenerator for creation of a CGNS file of the mesh, which is not shown in Fig. 5.2, since it is not an essential part needed for the Lattice Boltzmann Method. The main idea of this CGNS export is to use SamGenerator meshes in conventional CFD tools. Grid generation, especially in complex geometries, is a demanding task, which is handled automatically by SamGenerator, as it was demonstrated above. Since standard CFD tools rely on body conformal meshes, an extension of SamGenerator is currently under development, which modifies the boundary adjacent cells to fit to the boundary shape.

## 5.3 Solver - SamSolver

Before looking closer at the implementation of the Lattice Boltzmann Method in SamSolver, a general solution aspect of the Lattice Boltzmann equation, e.g., Eq. (3.16) for the SRT colli-

sion model, is addressed. For implementation, the LBE is decomposed into a transport and a collision step:

$$\text{Transport step : } \tilde{f}_i(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) \quad (5.1)$$

$$\begin{aligned} \text{Collision step : } f_i(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) = & \tilde{f}_i(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) \\ & + \Omega \left[ \tilde{f}_i^{eq}(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) - \tilde{f}_i(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) \right]. \end{aligned} \quad (5.2)$$

The transport step is a simple copy of a value from one position in memory to another position and thus free of round off errors. From Eq. (5.2) we see that the collision step uses only data of the node where it is executed. The resulting procedure for the LBM has therefore some beneficial features: Firstly, the scheme is explicit, there is no need to solve equations iteratively. Secondly, the collision step is purely local and the transport step involves only the lattice neighbors. This locality is the foundation for a highly efficient parallelization of the scheme.

Let us go into the details of SamSolver, which is currently supported on Windows and Linux operating systems. The Linux variant includes capabilities for usage and communication with the Platform LSF job scheduler, which is used as a workload management tool at the TU Kaiserslautern cluster. There are two versions of SamSolver for both operating systems, realizing different parallelization paradigms. The version for shared memory parallelism (SMP) uses the OpenMP API for code parallelization. For distributed memory parallelism, the message passing interface (MPI) is applied for the inter-process communication.

We will analyze the implementation structure of SamSolver by means of the flowchart of Fig. 5.3, which shows the basic conceptual subroutines.

### 5.3.1 Input Data

The first step of a SamSolver run is reading of input data. This starts with the input script, which holds all physical parameters, like boundary conditions and fluid properties, and control parameters, like the number of time steps to go in this run. Afterwards, the lattice definition is read from the sam file created by SamGenerator. In the case of MPI runs, the fluid domain must be decomposed into different fractions, which are calculated on the respective processors. For this decomposition task SamDecomposer has been developed. The output of SamDecomposer is one samd file per processor, which contains the decomposed part of the lattice and information about the inter-process communication. Depending on the version of SamSolver which is executed, the software will autonomously read the required sam or samd files from the specified input folder.

There is a further routine appendant to the input procedure, which is optional and thus not shown in Fig. 5.3. This routine reads the resume file of a previous run to resume a simulation.

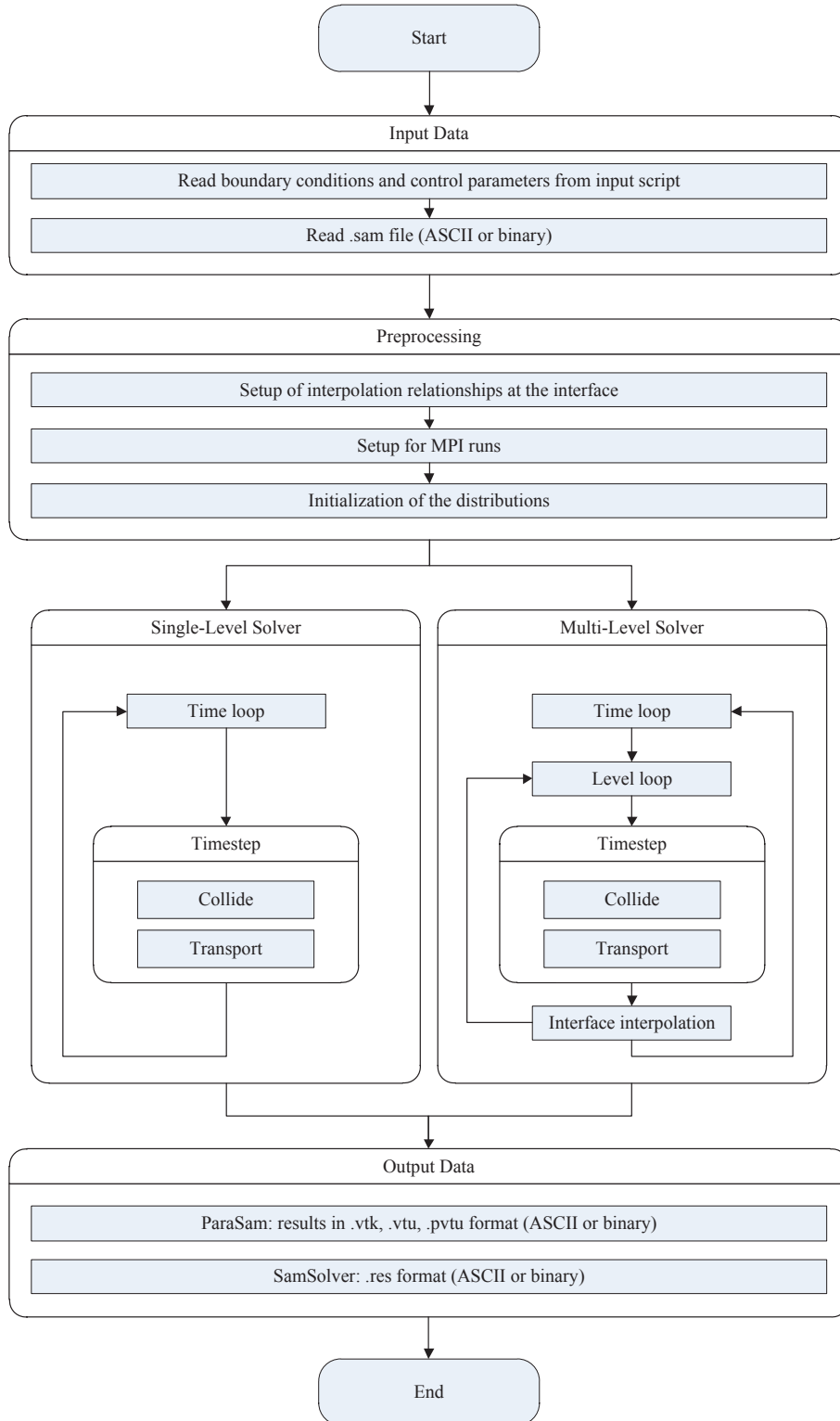


Figure 5.4: Flowchart of SamSolver

### 5.3.2 Preprocessing

After the lattice of the fluid domain is set up in the internal data structure of the SamSolver some preprocessing operations of administrative nature are necessary. From the interface definition in the sam file, the various relationships between the interface nodes, their partner nodes and the hanging nodes and their interpolation partners, need to be created for multi-level runs. In the case of unrefined grids we will refer to single-level and in the case of refined grids to multi-level runs in the sequel.

If the MPI version of SamSolver is running, a setup process for the inter-processor communication is mandatory. In this routine, the data which is send to and received from the other processes is determined and the according data structures are initialized.

The last step is the initialization of the distribution values for the numerical calculation. Initialization is done either by usage of the equilibrium distribution or by the iterative scheme of Mei et al. [87], as stated in Section 3.6. In both cases a global initialization velocity and density must be prescribed in the input script. In the case of a resumed simulation, the distribution values are initialized with its values of the previous run.

### 5.3.3 Solving

To reduce redundancies in the code and increase the ease of maintenance, a flexible way to execute different methods from the same call is essential. This is achieved in SamSolver by using C++ functors, which enable us to set the function to be executed during runtime. Depending on the run type of the simulation, i.e., single or multi-level, a specific solver is started, as depicted in Fig. 5.3.

The single-level solver consist of a time loop, in which the collision and transport steps are executed, until the specified number of time steps is reached. In principle, the sequence between collision and transport is interchangeable, but we experienced a better performance of the code in performing first collision and then transport. The call of the actual collision implementation is realized by a functor to permit the usage of the same code for SRT and MRT. Before the collision operation itself can be performed in the collision routine, the relaxation parameter must be calculated. For laminar calculations, the relaxation times are known a priori. In the case of turbulent flows, using an eddy viscosity based modeling,  $\nu_t$  is determined prior to the calculation of the relaxation parameter. The case of non-Newtonian fluids is similar to the turbulent one, as in each case the effective viscosity is varying. In SamSolver, non-Newtonian fluids are modeled by a power law approach, see Conrad [22] for an extensive introduction and validation of the non-Newtonian capabilities. The function call associated to the viscosity calculation is also performed by a functor to allow a great variety of different models.

The routine for the transport step includes the classic advection step for fluid nodes and also the handling of boundary conditions for boundary and wall nodes as described in Section 3.5. There are in principle two ways to carry out the transport step. Firstly, a node can push its distributions to its neighbors. Secondly, a node can pull the distributions from its neighbors. In



SamSolver we use the push approach, since it achieved a higher performance.

The structure of the multi-level solver in Fig. 5.3 is obviously more complex than the single level solver. Again, we have a time loop as most outer loop. Inside this time loop, we loop over the different lattice levels. This level loop includes, beside collision and transport routines, a routine for the interface interpolations, as explained in Section 3.7. Due to the nested time stepping (see Fig. 3.13), a clear sequence of the collision, transport, and interpolation actions on each particular level must be obeyed. An elegant way to implement this order by different loops was found by the developers and implemented in SamSolver; Conrad [22] explains this in more detail.

Another useful feature of SamSolver for batch calculations are the routines of the class SamCommandInterpreter. The class members of SamCommandInterpreter allow adjustments of different run parameters during execution. Communication with the solver is achieved by a script: the sam\_cmd file. In every iteration of the time loop, this script is checked for newly added keywords and according values. If new entries are present, they are checked for consistency and afterwards the associated solver run parameters are changed. For convenience, the capabilities of sending commands to the solver during runtime are included in the GUI SamLattice.

Of course, SamSolver and the remaining components of SAM-Lattice can be interpreted and used as a library itself. From this point of view, functors are not essential and the user can build its own application from the features of the library classes for every case. But as stated above, an aim of the code design is to create a tool, which can handle arbitrary geometries and flows, respectively, with a single application.

### 5.3.4 Output Data

For result evaluation with ParaSam, SamSolver includes the same output capabilities for vtk-compatible formats as SamGenerator. In MPI runs the output format is limited to pvtu, since this format is specifically designed for distributed calculations. The write interval as a number of time steps is a user input. During the time step loop, the solving routine will call the according output method, whenever a write interval is complete.

Besides the vtk-compatible output formats, SamSolver creates a resume (res) file at the end of every run. This res file is used to resume a simulation based on a completed run. The res file includes the distribution values for all directions at every node, which allows a smooth continuance of the simulation. Together with the sam file, the res file can be loaded in ParaSam, too. ParaSam then calculates the macroscopic moments for density, velocity, stress, and eddy viscosity from the distributions in the res file for result evaluation.

In addition to output of the complete flow field as vtk-compatible or res-file, SamSolver includes capabilities to monitor flow quantities which are written to a file and evaluated every time step. On the one hand, flow quantities at every node of the computational grid or at arbitrary points in the domain can be tracked. The fluid quantities at the nodes are directly calculated from the node's distributions, whereas values at monitor points are interpolated from

surrounding nodes by trilinear interpolation. On the other hand, SamSolver allows to record forces and torques on boundaries. Both quantities are calculated internally by the momentum exchange method [70] as introduced in Subsection 3.5.7. Monitoring of flow quantities, forces, and torques is very effective for usage as physical convergence criteria.

The output routines of SamSolver include some additional postprocessing capabilities which will be mentioned in the sequel. As we have seen, LBM is an intrinsic time-dependent scheme. But, oftentimes engineers are only interested in temporal averages of transient flows. To avoid excessive recording of transient results, solely for subsequent averaging, SamSolver has built-in averaging capabilities. The second feature is grid reduction. Especially for transient calculations where a huge amount of transient results is created, it can be necessary to reduce the memory requirements for output. With the grid reduction option,  $n^3$  cells,  $n \in [2, 3, \dots, 6]$ , are combined to a new coarse cell. This technique permits calculation on a fine grid and result storage on a coarser grid.

## CHAPTER 6

---

### Verification Cases

---

The development of a numerical scheme always includes verification and validation (V&V) processes. Purposes of these V&V processes are to benchmark the accuracy and suitability of a numerical approach to physical problems. The first step of this procedure is verification which is commonly subdivided into code and solution verification. Code verification aims to show that the mathematical model, on which the numerical scheme is based, is accurately solved. This means that the code is free of errors and bugs. In addition, solution verification turns out the numerical accuracy of the method by means of a certain calculation [3]. This includes the confirmation of the order of accuracy of the numerical scheme, as introduced in Subsection 3.8.1. For the identification of the solution error at best an exact solution of the verification case is used. When no exact solution is available, an error estimator can be used, as it is done, e.g., by Richardson extrapolation [34].

The definition of validation is given according to the ASME Standard for V&V in CFD [3]:

*“Validation is a process determining the degree to which a model is an accurate representation of the real world.”*

Validation is performed by comparison of simulation results with experimental measurements for a specified set of variables at certain conditions. A consequence of this practice is that there can be no validation without experimental data [3].

In this chapter the verification part of the V&V process will be performed for SAM-Lattice. By multiple benchmark cases, the different components and models of the CFD package will be investigated. As far as possible, the verification cases rely on exact solutions of the Navier-Stokes equations to avoid the use of error estimators. A reason for this is the fact that error estimators have certain requirements to be exact, like monotonic convergence in the case of Richardson extrapolation [34]. The solution error is expressed by the global absolute error to

the analytical solution for a particular flow quantity. This is exemplary expressed by Eq. (6.1) for the velocity component in x-direction, where the summation is performed over all  $N$  nodes of the grid:

$$E_u = \sqrt{\frac{1}{N} \sum_{\mathbf{x}} (u(\mathbf{x}) - u_{analytical}(\mathbf{x}))^2}. \quad (6.1)$$

## 6.1 Core Components

In this section, the basic components of SAM-Lattice will be verified. This means multiple laminar Newtonian flows, for which exact solutions of the Navier-Stokes equations exist, will be used for verification of the scheme including different kinds of boundary conditions.

### 6.1.1 Beltrami Flow

A known exact solution of the unsteady incompressible Navier-Stokes equation is Beltrami flow, which is characterized by a vorticity field parallel to the velocity. Furthermore, for Beltrami flows no term of the Navier-Stokes equations is zero. This is why they are often called full solutions of the Navier-Stokes equations. The best-known example of Beltrami flow is the 2D Taylor Green vortex flow. A detailed derivation of Beltrami flow and its solution is given by [32].

For benchmarking, we use here the Beltrami case proposed by [112], which has the following solution:

$$\begin{aligned} u &= -\frac{A}{k^2 + l^2} [\lambda l \cos(kx) \sin(ly) \sin(mz) + m k \sin(kx) \cos(ly) \cos(mz)] \exp(-\nu \lambda^2 t) \\ v &= \frac{A}{k^2 + l^2} [\lambda k \sin(kx) \cos(ly) \sin(mz) - m l \cos(kx) \sin(ly) \cos(mz)] \exp(-\nu \lambda^2 t) \\ w &= A \cos(kx) \cos(ly) \sin(mz) \exp(-\nu \lambda^2 t) \\ \text{with } \lambda^2 &= k^2 + l^2 + m^2. \end{aligned} \quad (6.2)$$

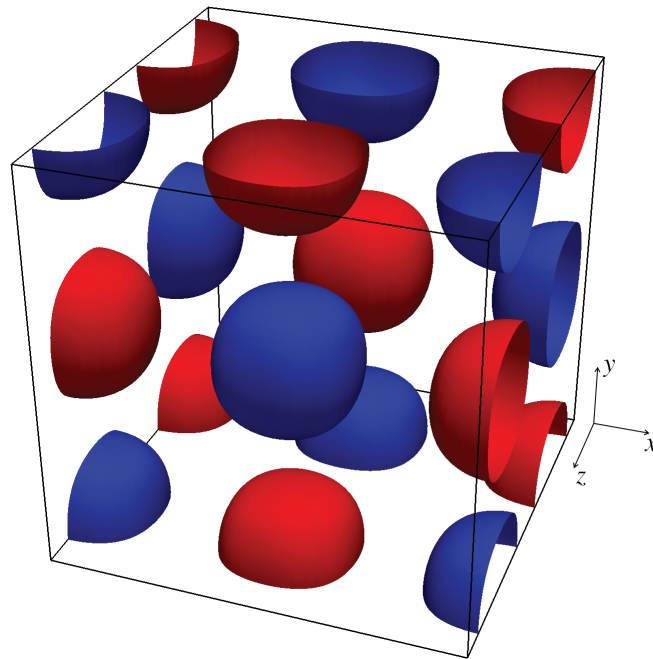
The flow described by Eq. (6.2) consists of a periodic array of viscously decaying counter-rotating vortices. The parameters  $k, l, m$ , which are wave numbers, control the number of vortices in each directions. For the calculations performed with SAM-Lattice, a cubic domain with an edge length of  $2\pi$  and periodic conditions on all sides is used. By this configuration, the basic algorithm can be tested without the influence of particular boundary conditions since periodic conditions do not alter the basic scheme. All flow parameters used for Beltrami flow are summarized in Table 6.1. With these given parameters, the maximum velocity  $|\mathbf{u}_{max}|$  in the domain at  $t = 0$  results to  $1 \text{ m/s}$ .

Table 6.1: Parameters for Beltrami Flow

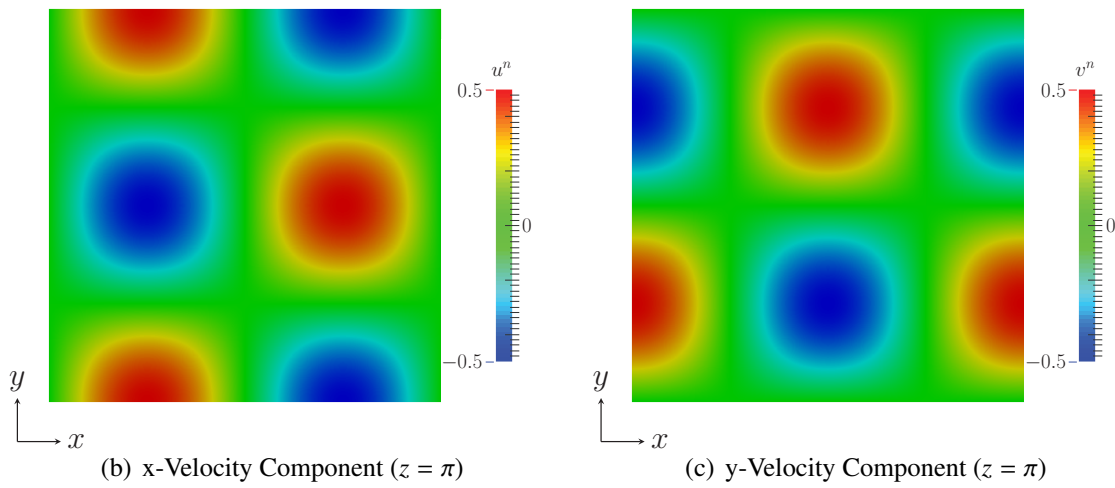
Parameter	Value	Unit
Domain size $L_x \cdot L_y \cdot L_z$	$2\pi \cdot 2\pi \cdot 2\pi$	$m^3$
Periodic directions	$x, y, z$	–
Amplitude $A$	1	–
Wave number $k, l, m$	1	–
Characteristic time $T$	$T = \frac{1}{\nu \lambda^2} = 3.333$	$s$
Characteristic length $L$	$L = \frac{L_x}{2\pi} = 1$	$m$
Maximum velocity $ \mathbf{u}_{max} $	1	$m/s$
Density $\rho$	1	$kg/m^3$
Kinematic viscosity $\nu$	0.1	$m^2/s$
Dimensionless velocity $u^n$	$u^n = \frac{u}{ \mathbf{u}_{max} }$	–
Dimensionless stress $\sigma_{yz}^n$	$\sigma_{yz}^n = \frac{\sigma_{yz}}{\rho  \mathbf{u}_{max} ^2}$	–
Dimensionless coordinate $x^n$	$x^n = \frac{x}{L}$	–
Spacing $\Delta x$	$2\pi/64$	$m$
Time step $\Delta t$	$T/500$	$s$
Reynolds number $Re$	$Re = \frac{ \mathbf{u}_{max} L}{\nu} = 10$	–
Lattice Mach number $Ma$	$Ma = \frac{ \mathbf{u}_{max} }{c_s} = 0.118$	–
Relaxation parameter $\Omega$	1.413	–
Free MRT relaxation parameter $s_i$	1.0	–

For visualization, the initial flow field at time  $t = 0$  is depicted in Fig. 6.1. In the top picture (Fig. 6.1(a)), the vortices are illustrated by isosurfaces of the  $z$ -velocity component. The two lower pictures show contour plots of  $u$  and  $v$  in a plane with constant  $z$ -coordinate, again the vortices are clearly visible. All quantities shown, are non-dimensional, what is indicated by the superscript  $n$ , now and in the following. Normalization is done with  $|\mathbf{u}_{max}|$ ; see Table 6.1.

The calculation for code verification is performed on a  $64^3$  cells grid. 500 time steps per characteristic time interval  $T$  are used for temporal resolution. Thus, the maximum lattice Mach number in the system is 0.118. The Beltrami flow is calculated with the SRT and MRT collision model. For the MRT calculation all free relaxation parameter  $s_i$  are set to 1.0, as it is always done throughout this thesis. Solution initialization is performed with the velocity field given by Eq. (6.2) for  $t = 0$  for the equilibrium part of the distributions. The non-equilibrium part



(a) Isosurfaces of z-Velocity Component ( $w^n = \pm 0.5$ )



(b) x-Velocity Component ( $z = \pi$ )

(c) y-Velocity Component ( $z = \pi$ )

Figure 6.1: Initialization Velocity Field ( $t = 0$ )

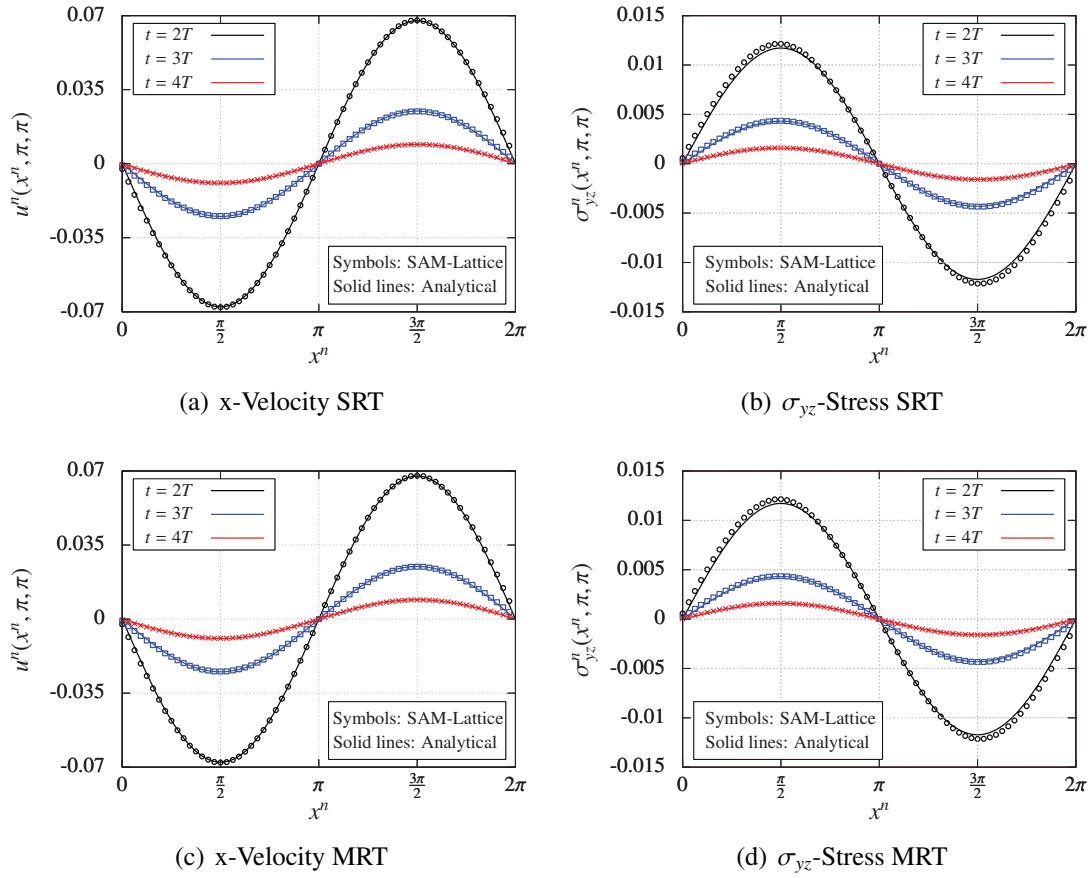


Figure 6.2: Comparison of Flow Quantities for Beltrami Flow

is initialized by the stress field which results from the derivatives of Eq. (6.2) and the relation (2.35). Calculations are conducted for a time period of  $32T$ . All numerical parameters are also included in Table 6.1 for a general overview.

Let us go on with a comparison of analytical and numerical results for code verification. Comparisons are given along a line at  $y = z = \pi$  for the normalized x-velocity component and the normalized  $\sigma_{yz}^n$  stress component.  $\sigma_{yz}^n$  is turned dimensionless by  $\rho |\mathbf{u}_{max}|^2$ . These two quantities have been chosen, since they are non-zero along the aforementioned line. Figure 6.2 shows multiple comparisons for the SRT and the MRT collision model at three different instants of time. The x-velocity reveals an excellent agreement between numerical and analytical results at all instants of time for the SRT (Fig. 6.2(a)) as well as for the MRT (Fig. 6.2(c)) model. Numerical results for  $\sigma_{yz}$  (Fig. 6.2(b) and 6.2(d)) are also in very good accordance with theoretical values at all instants, especially when we consider that the accurate calculation of stress requires a high spatial resolution.

Solution verification is done by a grid convergence study for the flow quantities  $u$  and  $\sigma_{yz}$ .

Table 6.2: Mesh Dependent Parameters for Beltrami Flow

Mesh size	$32^3$	$128^3$
Spacing $\Delta x$ [m]	$2\pi/32$	$2\pi/128$
Time step $\Delta t$ [s]	$T/125$	$T/2000$
Lattice Mach number $Ma$	0.236	0.059

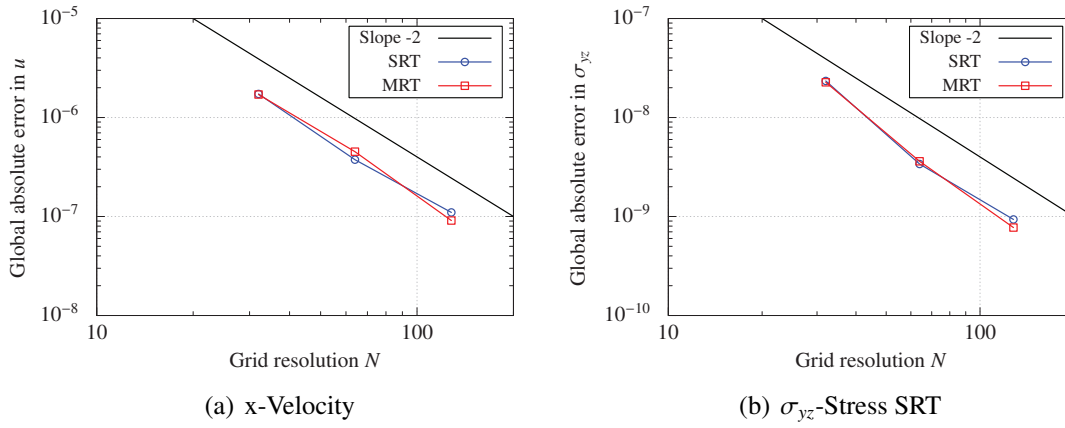


Figure 6.3: Grid Convergence for Beltrami Flow

Two further cases are considered for the convergence study, one coarser case with  $32^3$  cells and one finer case with  $128^3$  cells. The time steps on both grids are determined by diffusive scaling:  $\Delta t \propto \Delta x^2$ . The resulting time steps and lattice Mach numbers are given in Table 6.2, all other parameters are according to Table 6.1. Results for the absolute error in  $u$  at time  $t = 20T$  are given in double logarithmic diagram in Fig. 6.3(a). For comparison the diagram also shows the  $-2$  slope, which represents the theoretical convergence order. The errors for the SRT and MRT collision operators are nearly equal, but the SRT convergence rate reduces slightly from grid two to grid three. Nevertheless, both schemes manifest second order accuracy for  $u$ , the average convergence rate for the SRT model is 1.99 and for MRT is 2.1. Similar observations are taken from the convergence study for  $\sigma_{yz}$  in Fig. 6.3(b). Here, the average convergence rate for SRT is 2.32 and for MRT is 2.44.

These results for Betrami flow clearly verify the core implementation of the SRT and the MRT model. They also prove that the LBM scheme is second order accurate as theoretically derived in Chapter 3. Furthermore, we see from this unsteady case that LBM is feasible to recover the correct temporal evolution of the flow. In the subsequent sections the remaining components of SAM-Lattice will be investigated.



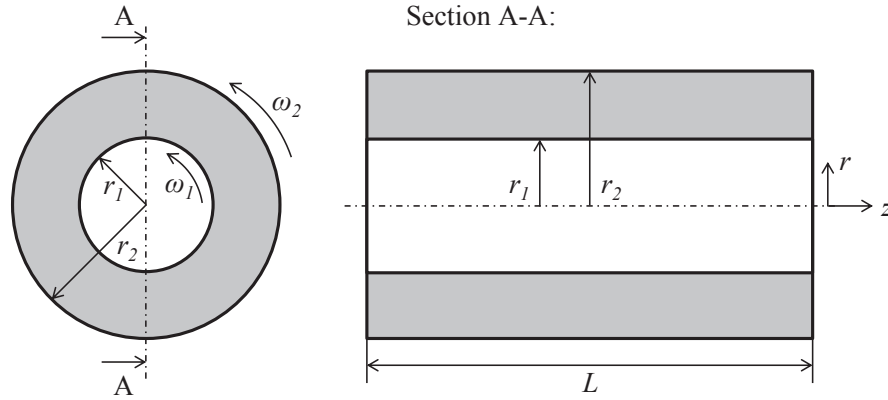


Figure 6.4: Sketch of Taylor-Couette Flow

### 6.1.2 Taylor-Couette Flow

The Navier-Stokes equations hold an exact solution for the flow in the annulus between two concentric rotating cylinders, which is known as Taylor-Couette flow and sketched in Fig. 6.4. Under certain assumptions, the analytical solution for the velocity field of Taylor-Couette flow is established in cylindrical coordinates in Eq. (6.3). These assumptions are: incompressible, steady and laminar flow, steady rotating cylinders, i.e.,  $\omega_1$  and  $\omega_2$  are constant, the absence of body forces and  $L \gg r_2$ ; see [106] for details.

$$\begin{aligned}
 u_z &= 0 \\
 u_r &= 0 \\
 u_\varphi(r) &= \frac{1}{r_2^2 - r_1^2} \left[ r(\omega_2 r_2^2 - \omega_1 r_1^2) - \frac{r_2^2 r_1^2}{r} (\omega_2 - \omega_1) \right]
 \end{aligned} \tag{6.3}$$

The Taylor-Couette flow has some technical applications, e.g., it is the basis of journal bearing flows. We will use the Taylor-Couette flow in this thesis as verification case for boundary schemes since the geometry contains complex boundaries. To definitely avoid a flow instability of toroidal vortices, which are known as Taylor vortices, only the outer cylinder is rotating and the inner one is standing still, see [106] for more details. In this case Eq. (6.3) reduces to:

$$u_\varphi(r) = \frac{1}{r_2^2 - r_1^2} \left[ r\omega_2 r_2^2 - \frac{r_2^2 r_1^2}{r} \omega_2 \right]. \tag{6.4}$$

Table 6.3: Parameters for Taylor-Couette Flow

Parameter	Value	Unit
Inner radius $r_1$	1	$m$
Outer radius $r_2$	2	$m$
Axial extension $L$	0.4	$m$
Periodic direction	$z$	–
Angular velocity $\omega_1$	0	$rad/s$
Angular velocity $\omega_2$	$5 \cdot 10^{-3}$	$rad/s$
Characteristic length $D_2$	$D_2 = 2 r_2 = 4$	$m$
Velocity at outer cylinder $u_2$	$u_2 = \omega_2 r_2 = 0.01$	$m/s$
Density $\rho$	1.225	$kg/m^3$
Kinematic viscosity $\nu$	$1.5 \cdot 10^{-3}$	$m^2/s$
Dimensionless velocity $u_\varphi^n$	$u_\varphi^n = \frac{u_\varphi}{u_2}$	–
Dimensionless radius $r^n$	$r^n = \frac{r}{r_2}$	–
Reynolds number $Re$	$Re = \frac{u_2 D_2}{\nu} = 26.67$	–
Cells in gap	80	–
Spacing $\Delta x$	0.0125	$m$
Time step $\Delta t$	0.009	$s$
Lattice Mach number $Ma$	$Ma = \frac{u_2}{c_s} = 0.0125$	–
Relaxation parameter $\Omega$	1.316	–
Free MRT relaxation parameter $s_i$	1.0	–

All parameters used for the benchmark calculations with SAM-Lattice are listed in Table 6.3. The computational domain is set periodically in axial direction to obey  $L \gg r_2$  and to simultaneously reduce the computational domain size. Since this is a steady case, simulations are run until steady state is reached. Steady state is assumed if the error defined by Eq. (6.1) is changing less than  $10^{-12}$  per time step. Then the simulations are stopped. This criterion will also be applied to the following steady flow calculations. For initialization of the distributions the equilibrium distribution with a zero velocity is used.

Calculations are performed with the SRT and MRT model, using the boundary conditions according to Bouzidi and Mei at the inner and outer cylinder, respectively. The velocity of the

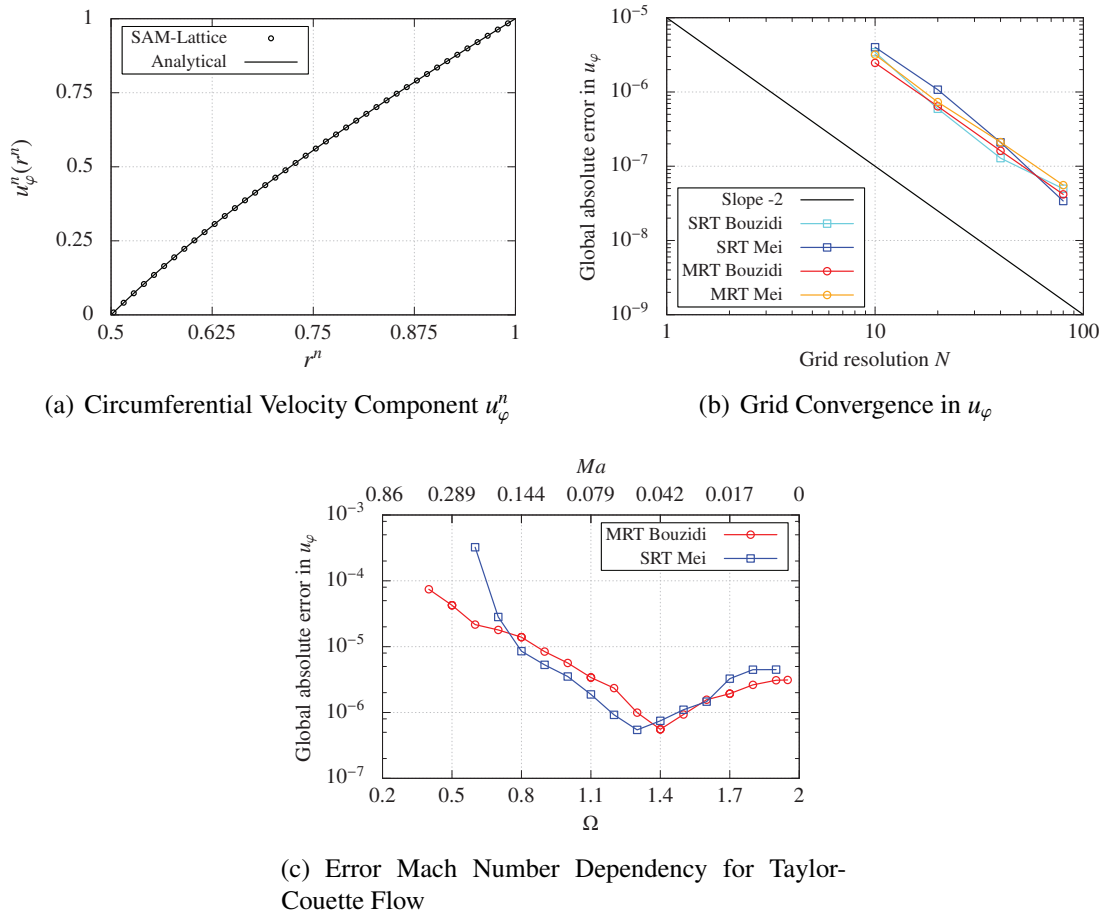


Figure 6.5: Results for Taylor-Couette Flow

outer cylinder is determined from the angular velocity  $u_2 = \omega \times r_2$  and applied in the according scheme. For code verification, a comparison between the analytical solution for  $u_\varphi$  and the SAM-Solver results is shown in Fig. 6.5(a) for the MRT Bouzidi case. The velocity  $u_\varphi^n$  is turned dimensionless by  $u_2$  and, for better clarity, only every second point of the numerical solution is depicted. We see a perfect match between the analytical and numerical results. A comparison for the three remaining cases will be omitted here since they show the same excellent qualitative behavior. Instead, we will have a look at the grid convergence of the different boundary schemes.

Three coarser additional grids are used for the grid convergence study. Again the diffusive scaling is applied for time step determination. Parameters, which change for the additional grids, are listed in Table 6.4.

The results of the grid convergence study are drawn in Fig. 6.5(b). All error curves are in a comparable range, there is no distinct favorite resulting from the combination of the different

Table 6.4: Mesh Dependent Parameters for Taylor-Couette Flow

Cells in annulus	10	20	40
Spacing $\Delta x$ [m]	0.1	0.05	0.025
Time step $\Delta t$ [s]	0.576	0.144	0.036
Lattice Mach number $Ma$	0.1	0.05	0.025

Table 6.5: Average Grid Convergence Rate for Taylor-Couette Flow

Case	SRT Bouzidi	SRT Mei	MRT Bouzidi	MRT Mei
Average grid convergence in $u_\varphi$	2.036	2.293	1.96	1.947

boundary schemes with the collision models. From the average convergence rate, listed in Table 6.5, the combination SRT Mei has the highest rate, but obviously the convergence rates of all schemes are close to 2. This reveals that the advanced bounce back schemes are able to retain the second order convergence of the method. The convergence rate of the MRT scheme can be tuned by optimizing the free relaxation parameters  $s_i$ , but this will not be shown here.

Another very interesting point for steady state flows is the choice of the time step. In unsteady flows the maximum time step is limited, due to the flow physics. In contrast, for steady flows one tends to use large time steps to reach the steady state fast. As we know from Eqs. (3.5), (3.25) and (3.29) the time step is directly linked to the lattice Mach number. Thus, increasing  $\Delta t$  will increase  $Ma$ , as we have experienced from the diffusive scaling as well. For a fixed grid spacing, changes in  $\Delta t$  also influence the value of the collision parameter  $\Omega$ . When dealing with (weakly) compressible flows, one is often willing to match the physical  $Ma$  in the lattice as well. But for incompressible flows, the lattice Mach number is an artificial quantity which influences the solution via the compressibility error  $E_{Ma}$ . From Eq. (3.81), we expect that  $E_{Ma}$  decreases quadratically with  $\Delta t$  on a fixed grid. In the same way, the overall error reduces until it converges to the spatial discretization error  $E_x$ , when  $E_x$  gets dominant. To numerically investigate the behavior of the overall error, Taylor-Couette flow, as defined above, is calculated on mesh2, for different  $Ma$  or  $\Delta t$ , respectively. The results of this parameter study are shown for two cases, SRT Mei and MRT Bouzidi, in Fig. 6.5(c). Here, the global absolute error is drawn against  $Ma$  on the top abscissa and against  $\Omega$  on the bottom abscissa, respectively.

For both cases, the absolute error first decreases with  $Ma$  until a minimum is reached. The minimum position differs in the SRT and MRT case. The results of [22] show that the location of the minimum can be varied in the MRT case by changing the free relaxation parameters. After the minimum, the error is rising again, what is contrary to Eq. (3.81), which predicts a converging error in the incompressible limit,  $Ma \rightarrow 0$ . These observations can also be found in literature [27, 56]. A possible explanation for this behavior is given by [56], who show that the

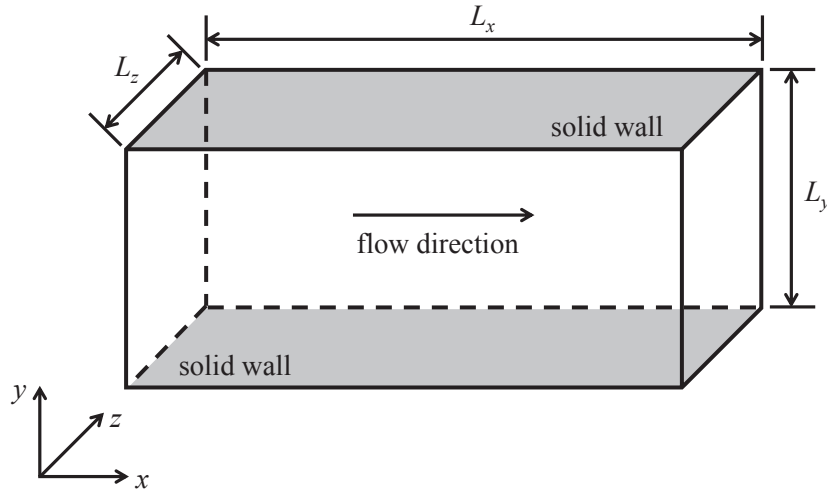


Figure 6.6: Domain for Channel Flow

LBM scheme can be expressed on the basis of equilibrium distributions only and the effective stencil needed therefor depends on the relaxation parameter  $\Omega$ . Hence,  $\Omega$  directly influences the scheme and there is a specific value of  $\Omega$  for which the lowest error occurs. The quintessence of this result is that smaller errors in incompressible flows are not necessarily achieved by reducing the lattice Mach number. Instead, experience shows that the smallest errors usually occur in a range of  $\Omega$  between 1.0 – 1.4.

### 6.1.3 Couette Flow

In this subsection the accuracy of the shear boundary conditions will be checked. For this purpose the well known Couette flow is investigated. As a first test case, the Couette flow in a plane channel will be simulated, see Fig. 6.6. In the Couette flow, one of the channel walls is moving with a certain velocity, while the other one is at rest. In our fictitious test case, we assume that the upper wall is moving with  $u_{top}$  in x-direction and at the lower wall a certain slip appears. This slip is characterized by a specified shear  $\tau_{bottom}$ . An analytical solution of the Navier-Stokes equations with these boundary values can be achieved with certain additional assumptions. These are based on steady, laminar flow in an infinite plane channel without pressure gradients and external forces, see [106] for details. The Navier-Stokes equations simplify in this case to:

$$\frac{d^2 u}{dy^2} = 0, \quad (6.5)$$

Table 6.6: Parameters for Plane Couette Flow

Parameter	Value	Unit
Half channel height $H$	1	$m$
Domain size $L_x \cdot L_y \cdot L_z$	$H \cdot 2H \cdot 2H$	$m^3$
Periodic directions	$x, z$	–
Density $\rho$	1	$kg/m^3$
Kinematic viscosity $\nu$	1	$m^2/s$
Velocity top wall $u_{top}$	3	$m/s$
Shear at bottom wall $\tau_{bottom}$	1	$Pa$
Dimensionless velocity $u^n$	$u^n = \frac{u}{u_{top}}$	–
Dimensionless coordinate $y^n$	$y^n = \frac{y}{L_y}$	–
Reynolds number $Re$	$Re = \frac{u_{top} 2H}{\nu} = 6$	–
Cells per $L_y$	20	–
Spacing $\Delta x$	0.1	$m$
Time step $\Delta t$	$9.62 \cdot 10^{-4}$	$s$
Lattice Mach number $Ma$	$Ma = \frac{u_{top}}{c_s} = 0.05$	–
Relaxation parameter $\Omega$	1.268	–
Free MRT relaxation parameter $s_i$	1.0	–

where the axis directions of Fig. 6.6 are used. If we assume that the origin of the coordinate system is at the bottom wall, the velocity profile is achieved from Eq. (6.5) with the boundary values:

$$\begin{aligned}
 u(y = L_y) &= u_{top} & \left. \frac{du}{dy} \right|_{y=0} &= \frac{\tau_{bottom}}{\mu} \\
 \Rightarrow u(y) &= \frac{\tau_{bottom}}{\mu} (y - L_y) + u_{top}.
 \end{aligned} \tag{6.6}$$

In the simulations, the domain, as shown in Fig. 6.6, is set periodically in  $x$  and  $z$ -direction to reproduce an infinite expansion. The domain size is  $H \times 2H \times 2H$ , where  $H$  is half of the channel height. To model the top wall, the bounce back rule, Eq. (3.45), is used. For the bottom wall the shear bounce back (SBB) or the shear anti bounce back (SABB) schemes, as introduced in Subsection 3.5.7, are applied. Once more, initialization is performed with an equilibrium of zero velocity. Table 6.6 arranges the simulation parameters.

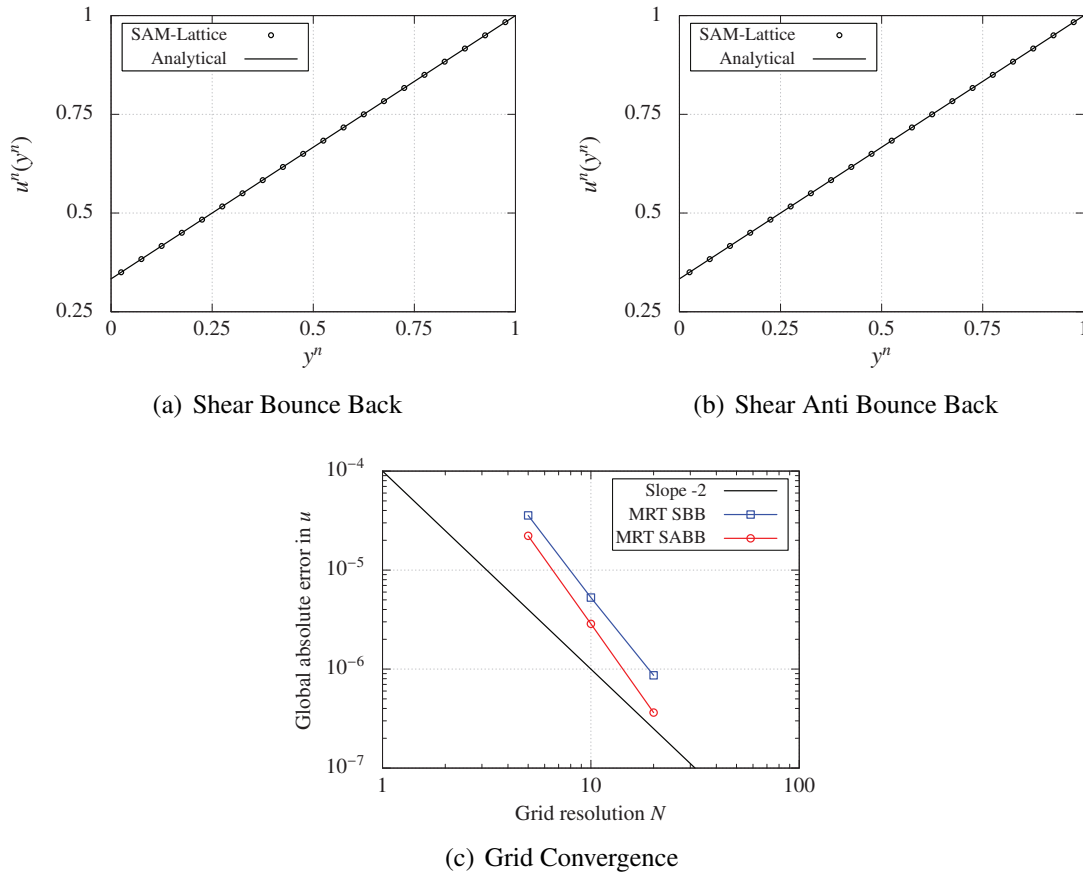


Figure 6.7: Results for Plane Couette Flow

Since the bounce back rule is used, second order accuracy is only achieved if the top wall is located half a spacing away from the first fluid node. This fact is considered during mesh creation and the top and bottom wall are placed half a spacing apart from the adjacent fluid nodes. Here, only the MRT collision model is considered, but the results are transferable to SRT as well.

The streamwise velocity profile of the simulation for SBB and SABB are compared to the analytical solution in non-dimensional formulation in Figs. 6.7(a) and 6.7(b). Normalization is performed by  $u_{top}$  for the ordinate and the channel height  $L_y = 2H$  for the abscissa. We see an excellent agreement between theoretical and numerical results for both shear boundary conditions. Next, a grid convergence study for both SBB and SABB is performed. Two further meshes are constructed for this investigation, where the grid spacing is once doubled and once halved compared to Table 6.6. Diffusive scaling is applied for the determination of the time steps on the different grids. The parameters which differ to the primary mesh are listed in Table 6.7.

Table 6.7: Mesh Dependent Parameters for Plane Couette Flow

Cells per $L_y$	10	20
Spacing $\Delta x$ [m]	0.2	0.05
Time step $\Delta t$ [s]	$3.848 \cdot 10^{-3}$	$2.405 \cdot 10^{-4}$
Lattice Mach number $Ma$	0.1	0.025

From the results of the convergence study in Fig. 6.7(c), we see that SBB and SABB errors are about the same order of magnitude and second order spatial convergence is achieved. The average convergence rate for SABB is about 2.9 and a bit higher than for SBB with 2.7. This proves that second order convergence is attained with SBB and SABB for axis aligned walls located half a spacing off the first fluid nodes. Besides, these results also verify the standard bounce back.

For verification of the shear schemes in complex domains, the Couette flow is used again, but in the annulus domain, according to Fig. 6.4. In contrast to Taylor-Couette flow, the outer cylinder is not rotating now, instead it is sliding in axial direction. If we assume once more incompressible, steady, laminar flow, the absence of body forces and pressure gradients, and  $L \gg r_2$ , an analytical solution of the Navier-Stokes equations is achieved for this type of flow, see [130]. The only non-zero velocity component occurs in axial direction and is described by:

$$u_z = C_1 \ln(r) + C_2. \quad (6.7)$$

The constants  $C_1$  and  $C_2$  are determined from the boundary conditions of our fictitious test case. We assume a sliding outer cylinder in z-direction  $u_z(r_2) = u_{oc}$  and a certain slip at the inner cylinder  $\tau_w(r_1) = \tau_{ic}$ . Finally, the velocity distribution reads as:

$$u_z = \frac{\tau_{ic}}{\mu} r_1 \ln\left(\frac{r}{r_2}\right) + u_{oc}. \quad (6.8)$$

Since the domain is the same as for Taylor-Couette flow, many simulation parameters can be taken from Table 6.3. Table 6.8 shows only the parameters which are different to Taylor-Couette flow. Wall modeling of the outer cylinder is performed by the Bouzidi scheme. SBB and SABB are used at the inner cylinder. Again, the distributions are initialized with an equilibrium resulting from zero velocity. Only the SRT collision model is applied for this case, but results are transferable to MRT.

Figures 6.8(a) and 6.8(b) contrast the simulation results of the SBB and SABB schemes with the analytical solution in a dimensionless representation. An excellent match for both shear schemes is also achieved for the more complex case. The remaining graph, Fig. 6.8(c), depicts the outcome of a grid convergence study. For the grid convergence study three additional meshes are used and the mesh specific parameters are listed in Table 6.9. The results of



Table 6.8: Parameters for Annulus Couette Flow

Parameter	Value	Unit
Angular velocity $\omega_1$	0	<i>rad/s</i>
Angular velocity $\omega_2$	0	<i>rad/s</i>
Axial velocity of outer cylinder $u_{oc}$	3	<i>m/s</i>
Shear at inner cylinder $\tau_{ic}$	1	<i>Pa</i>
Density $\rho$	1	<i>kg/m<sup>3</sup></i>
Kinematic viscosity $\nu$	1	<i>m<sup>2</sup>/s</i>
Dimensionless velocity $u_z^n$	$u_z^n = \frac{u_z}{u_{oc}}$	–
Reynolds number $Re$	$Re = \frac{u_{oc} D_2}{\nu} = 12$	–
Cells in gap	40	–
Spacing $\Delta x$	0.025	<i>m</i>
Time step $\Delta t$	$1.2 \cdot 10^{-4}$	<i>s</i>
Lattice Mach number $Ma$	$Ma = \frac{u_{oc}}{c_s} = 0.025$	–
Relaxation parameter $\Omega$	1.02	–

Table 6.9: Mesh Dependent Parameters for Annulus Couette Flow

Cells in annulus	10	20	80
Spacing $\Delta x$ [ <i>m</i> ]	0.1	0.05	0.0125
Time step $\Delta t$ [ <i>s</i> ]	$1.92 \cdot 10^{-3}$	$4.8 \cdot 10^{-4}$	$3.0 \cdot 10^{-5}$
Lattice Mach number $Ma$	0.1	0.05	0.0125

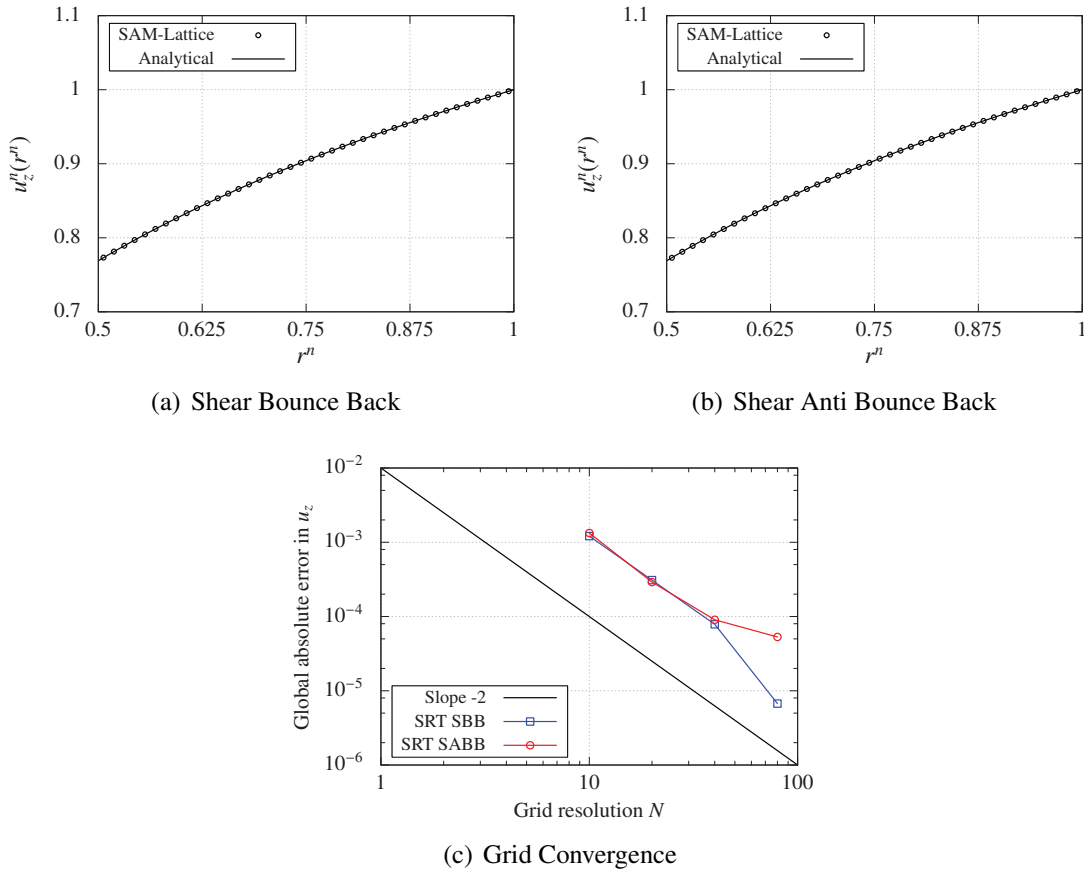


Figure 6.8: Results for Annulus Couette Flow

convergence study reveal that absolute errors of SBB and SABB on the first three meshes are quite equal, afterwards the SABB error is decreasing less than SBB. This is provoked by the convergence behavior. The SBB scheme shows the expected second order convergence rate, which comes to 2.4 in average. At first, the SABB's convergence rate is second order, too. But, when further refining the mesh, the boundary error gets dominant and the convergence rate reduces to first order. This is in accordance with the observations for the volumetric definition of the scheme, which also offers first order convergence behavior [77].

In summary, the findings of this subsection verify the two proposed shear boundary conditions in simple and complex domains.

### 6.1.4 Jeffery-Hamel Flow

To close the verification process of the core components, the grid refinement techniques needs to be tested. This is done with the flow in a diverging channel as a benchmark case, see Fig.

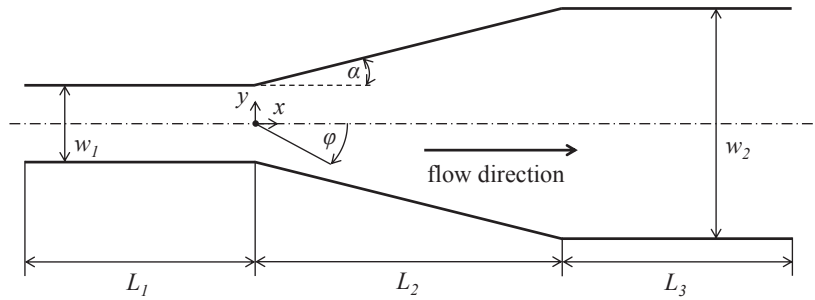


Figure 6.9: Domain for Jeffery-Hamel Flow

6.9. For fully developed flows in converging and diverging plane channels Jeffery and Hamel provided an exact similarity solution of the Navier-Stokes equations. This similarity solution results in an ordinary differential equation (ODE) for the velocity [130]. For the solution of the ODE we use the variation of parameter method, as introduced in [65] for Jeffery-Hamel flows. By applying this technique, an iterative scheme to calculate the velocity is derived, see [65] for details. The calculation of our reference solution is performed by this iterative procedure with the commercial software Mathematica.

Let us now introduce the case to be investigated and the numerical model used therefor. In Fig. 6.9 we see that beside the actual diffuser an inlet and outlet zone is present. The sketch of the geometry shows only a sectional view of the domain, since the domain is periodic in  $z$ -direction to create an infinitely large channel. Table 6.10 lists the geometrical dimensions of all parts of this test case. An additional inlet and outlet zone is necessary due to the applied boundary conditions. At the inlet a constant velocity is specified according to the investigated flow rate of  $Q = 5.338 \text{ m}^3/\text{h}$ . At the outlet a constant gauge pressure of  $p_{out} = 0 \text{ Pa}$  is prescribed. Through the usage of constant values along the boundary, the flow needs some distance to develop correct flow profiles. Within the Lattice Boltzmann model, the top and bottom wall of the channel are modeled with Bouzidi's scheme. An equilibrium distribution build with the gauge pressure and an extrapolated velocity is used at the outlet. The Dirichlet condition at the inlet is again realized with the Bouzidi scheme and a periodic condition was applied in  $z$ -direction. All parameters of the numerical model are summed up in Table 6.10, too.

To test the refinement techniques, the grid is refined twice towards the upper and lower channel wall, since high gradients are present in these regions. Both refinement levels contain at least 6 cells, what is achieved by using the boundary related refinement technique. Due to the different scaling procedures for SRT and MRT, calculations with both collision operators are performed for verification. All distributions are initialized with an equilibrium of zero velocity.

Table 6.10: Parameters for Jeffery Hamel Flow

Parameter	Value	Unit
Length inlet zone $L_1$	0.6	$m$
Length diffuser $L_2$	0.8	$m$
Length outlet zone $L_3$	0.3	$m$
Half opening angle $\alpha$	5	$^\circ$
Extension in z-direction $L_z$	0.08	$m$
Periodic direction	$z$	–
Density $\rho$	1.225	$kg/m^3$
Kinematic viscosity $\nu$	$1.39 \cdot 10^{-5}$	$m^2/s$
Flow rate $Q$	5.338	$m^3/h$
Gauge pressure at domain outlet $p_{out}$	0	$Pa$
Dimensionless velocity $u^n$	$u^n = \frac{u}{u_{max}}$	–
Dimensionless angle position $\varphi^n$	$\varphi^n = \frac{\varphi}{\alpha}$	–
Reynolds number $Re$	$Re = \frac{u_{max} r \alpha}{\nu} = 10$	–
Spacing in level 0 $\Delta x_0$	0.04	$m$
Time step in level 0 $\Delta t_0$	$1.66 \cdot 10^{-2}$	$s$
Lattice Mach number $Ma$	$Ma = \frac{u_{max}}{c_s} = 0.1$	–
Relaxation parameter in level 0 $\Omega_0$	1.8406	–
Free MRT relaxation parameter $s_i$	1.0	–

The velocity profiles of the diffuser flow are evaluated at three different x-positions of 0.2, 0.4 and 0.6  $m$ . A result comparison of LBM and the reference solution at the different positions is depicted in Fig. 6.10 by means of dimensionless parameters. Additionally to the velocity profiles, the mesh at the according position is drawn in the graphic. The grid spacing is constant at all positions. The optical change in the different subfigures is due to the normalization and the different channel heights at the multiple diffuser positions. All plots for MRT and SRT offer a perfect accordance between numerical and reference solution at every position. At the grid interfaces, which are highlighted by the dashed vertical lines, smooth transitions are observed, i.e., there are no discontinuities detectable.

An error calculation is omitted here, since there is no closed analytical form of the reference

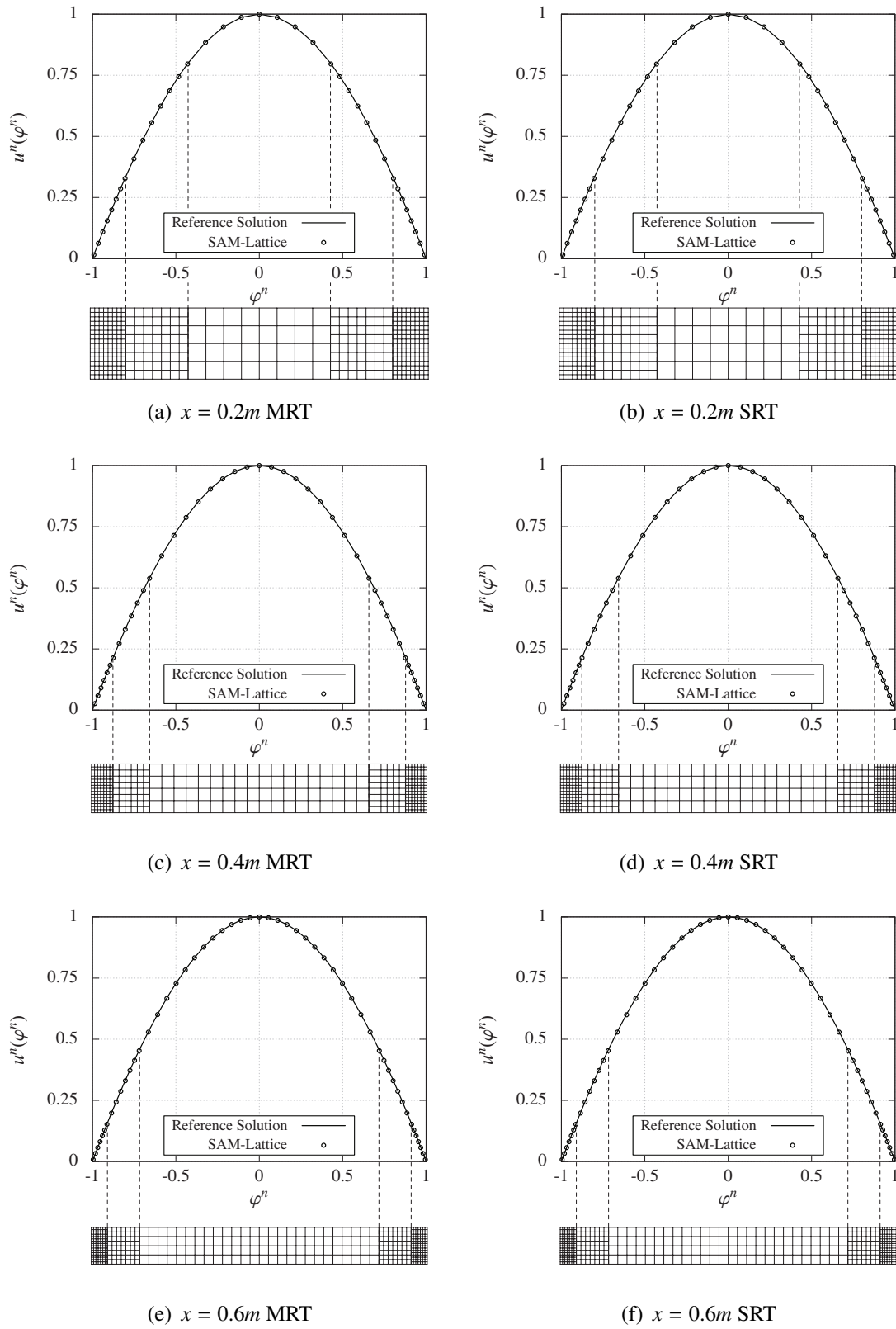


Figure 6.10: Results for Jeffery-Hamel Flow

solution, which is instead achieved by a numerical process as well. The outcomes of the calculation for Jeffery-Hamel flow verify the implemented grid refinements techniques for SRT as well as for MRT.

## 6.2 LES Components

After the successful verification of SAM-Lattice's core components, we will now test the large eddy model, including the associated boundary conditions. The structure of this section is orientated on the different boundary treatments, rather than on different flow types, as it is done in the previous section. This change is motivated by the fact that the LES features will mainly be tested for turbulent channel flow, since a very good DNS database is available for this case.

### 6.2.1 Wall Resolved Approach

The LES-LBM concept in combination the Smagorinsky model is applied to turbulent channel flow for verification. First, the wall resolved approach is used, since no additional wall modeling is adopted and the focus can be turned on the performance of the LES concept itself. We will start with a single level grid and a channel flow for a friction Reynolds number  $Re_\tau = 180$ , what corresponds to a Reynolds number  $Re_c \approx 4200$ , build with the according centerline velocity. The flow domain is equal to Fig. 6.6 and periodic in x and z direction, but the domain size is now  $4H \cdot 2H \cdot 2H$  according to the minimum specifications of [62]. The flow setup is readily explained: The two channel walls are handled by the standard bounce back. This implies that the first fluid nodes off the wall are located  $0.5\Delta x$  away from the wall and this is considered in grid generation. Furthermore, to ensure an accurate boundary layer resolution the first fluid nodes must fulfill  $y^+ \approx 1$ . The flow is driven by a uniform body force, which correlates to the wall shear and the friction velocity, respectively, by:

$$\mathbf{f} = \rho \mathbf{a} = -\frac{dp}{dx} \mathbf{x} = \frac{\tau_w}{H} \mathbf{x} = \frac{\rho u_\tau^2}{H} \mathbf{x}. \quad (6.9)$$

As said before, Smagorinsky's model (Eq. (4.12)) is used as subgrid model within the LES methodology, see Chapter 4. The Smagorinsky constant  $C_s$  is chosen to 0.1. Due to the required wall resolution  $y^+ \approx 1$  and the uniform grid, the dimensionless spacing results in  $\Delta x^+ = \Delta y^+ = \Delta z^+ \approx 2$ . This might be a quite fine resolution for a large eddy simulation, but according to the numerical model, the LES should reduce to DNS if all scales are resolved. The MRT collision model is applied, since SRT suffers from numerical stability problems and spurious oscillations in this case. All distribution values are initialized with a zero velocity equilibrium. In the course of the simulation the flow is accelerated by the body force and, as a result, a turbulent flow is established. Evaluation of results is started after a statistically steady state of the turbulent flow has been reached and performed over a period of  $14H/u_\tau$ . For convenience, all parameters of the domain and the simulation setup are summarized in Table

Table 6.11: Parameters for Turbulent Channel Flow at  $Re_\tau = 180$ 

Parameter	Value	Unit
Half channel height $H$	1	$m$
Domain size $L_x \cdot L_y \cdot L_z$	$4H \cdot 2H \cdot 2H$	$m^3$
Periodic directions	$x, z$	–
Density $\rho$	1.225	$kg/m^3$
Kinematic viscosity $\nu$	$1.39 \cdot 10^{-5}$	$m^2/s$
Friction Reynolds number $Re_\tau$	$Re_\tau = \frac{u_\tau H}{\nu} = 180$	–
Friction velocity $u_\tau$	$2.502 \cdot 10^{-3}$	$m/s$
Acceleration $ \mathbf{a} $	$6.26 \cdot 10^{-6}$	$m/s^2$
Dimensionless turbulent kinetic energy $E^n(\kappa)$	$E^n(\kappa) = \frac{E(\kappa)}{\sum_\kappa E(\kappa)}$	–
Dimensionless wave number $\kappa^n$	$\kappa^n = \frac{2\pi f}{u_{max}} H$	–
Cells per $H$	90	–
Spacing $\Delta x$	0.0111	$m$
Time step $\Delta t$	$1.07 \cdot 10^{-3}$	$s$
Lattice Mach number $Ma$	$Ma = \frac{24 \cdot u_\tau}{c_s} = 0.01$	–
Smagorinsky constant $C_s$	0.1	–
Free MRT relaxation parameter $s_i$	1.0	–
Averaging period $\Delta T$	$14 \frac{H}{u_\tau}$	$s$

6.11. The relaxation parameter  $\Omega$  is not listed in the table since it is a local quantity depending on the eddy viscosity  $\nu_t$ .

Figure 6.11 shows an instantaneous snapshot of the flow field by means of  $u^+$  (Eq. (4.25)) for an overview. The turbulent nature of the flow is clearly visible. The next graphic in Fig. 6.12 depicts the mean velocity profile by means of  $u^+$  and  $y^+$ . For comparison, the DNS results of Moser et al. [88] are used as a reference solution. We see a very good match between the LES-LBM results and the DNS data in all layers of the velocity profile. The turbulent fluctuations  $u'^+$ ,  $v'^+$  and  $w'^+$  are drawn in Fig. 6.13. Again, the SAM-Lattice results agree very well with the DNS solution. From the turbulent fluctuations we see that the LES is well resolved and most of the turbulent fluctuations are represented, due to the high resolution. This is also expressed in the ratio of  $\nu_t/\nu$ . It will not be shown here, but it is below 1 in the complete domain.

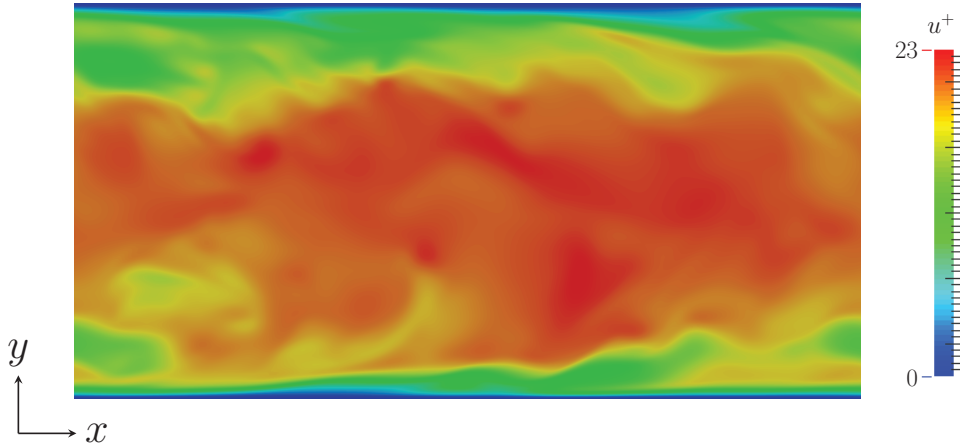


Figure 6.11: Instantaneous Velocity Field

To conclude this simulation, the spectrum of the turbulent kinetic energy is investigated. The spectrum has been analyzed at a position  $y^+ \approx 41$  and calculated by a Fourier transformation of the velocity fluctuation as a function of time using Taylor's hypothesis, see, e.g., [94] for details. Figure 6.14 shows the normalized turbulent kinetic energy versus the normalized wave number. Normalization of kinetic energy was performed by the total resolved kinetic energy and for the wave number by the half channel height  $H$ . The wave number itself was calculated from the according frequency by the following dispersion relation:

$$\kappa = \frac{2\pi f}{u_{max}}, \quad (6.10)$$

where  $u_{max}$  is the average centerline velocity. The normalization of the turbulent energy in the mentioned way allows us to see directly which percentage of the total energy is contained by a certain wave number or a range, respectively. Commonly, the energy spectra are subdivided into three regions [94]: The energy containing range, the inertial range and the dissipation range. The energy containing subrange is formed from the large scales and extends up to a dimensionless wave number  $\kappa'' \approx 1$ . Afterwards the inertial subrange starts and reaches to  $\kappa'' \approx 20$ . Characteristic for the inertial subrange is a slope of  $-5/3$  in the log-log plot. This slope is well reproduced by the simulation, as a comparison with the reference line in the diagram illustrates. Thereafter, the dissipation range begins and extends to the cutoff wave number of the filter. The cutoff wave number of the numerical grid is in dimensionless description  $\kappa'' = \frac{\pi}{\Delta x}H = 282.74$ . From Fig. 6.14, we see that the effective cutoff wave number is about 150 and reveals the difference to the analytical filter, as explained in Section 4.1.

Since there is no grid independency in LES, due to the facts stated above, no convergence studies will be performed here. The results of the presented turbulent channel flow clearly verify the LES-LBM scheme implemented in SAM-Lattice for single-level grids. Next, we will perform verification for the multi-level scheme.



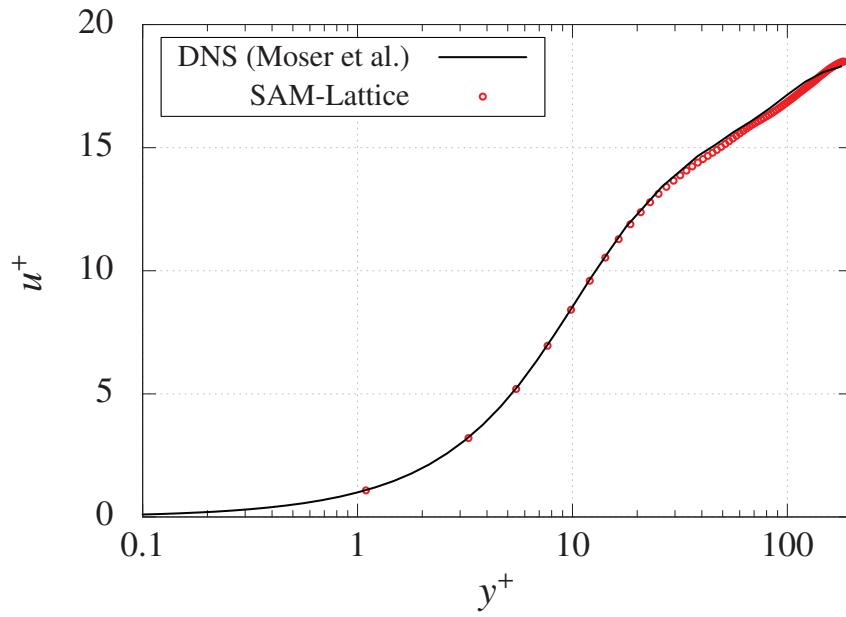


Figure 6.12: Mean Velocity Profile

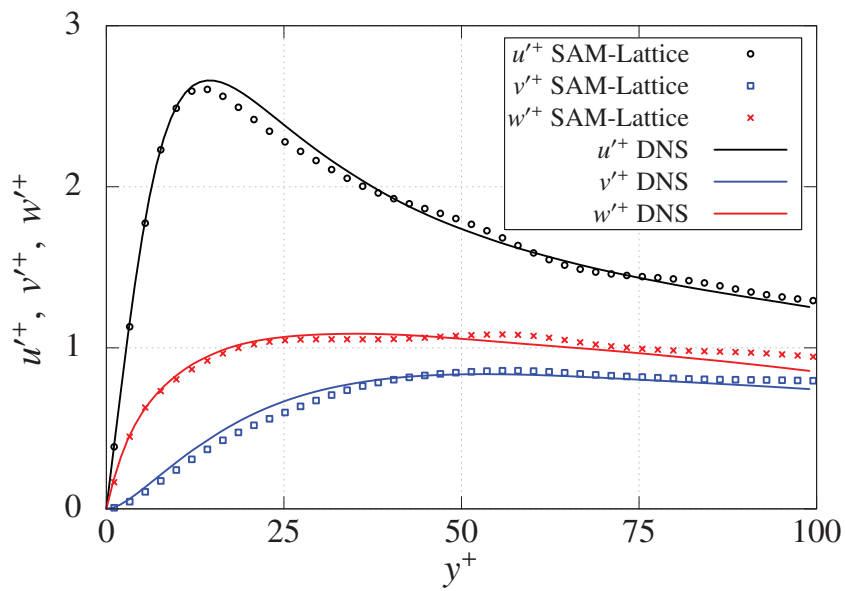


Figure 6.13: Turbulent Velocity Fluctuations

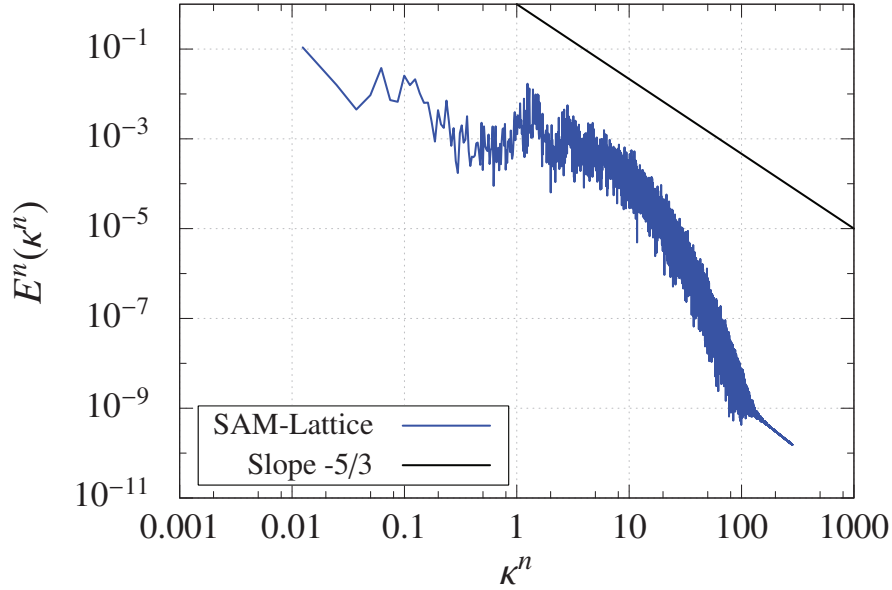


Figure 6.14: Turbulent Kinetic Energy Spectrum

Table 6.12: Parameters for Turbulent Channel Flow at  $Re_\tau = 180$  with Grid Refinement

Parameter	Value	Unit
Cells per $H$	45	–
Spacing in level 0 $\Delta x_0$	0.0222	$m$
Time step in level 0 $\Delta t_0$	$2.14 \cdot 10^{-3}$	$s$

Again, turbulent channel flow at  $Re_\tau = 180$  is used as benchmark case for the LES-LBM multi-level approach. Hence, we can keep it short and list only the differences of the setup compared to the single-level case in Table 6.12. The grid spacing  $\Delta x_0$  in level 0 is doubled compared to the single-level case and the grid is once refined locally from the channel walls up to  $y^+ \approx 43$ . Due to stability reasons only the MRT model is applied.

Let us go on with the results for the multi-level run, which hold the same presentations as for the single-level run to allow direct comparisons. The instantaneous flow field in Fig. 6.15 visualizes the turbulent character of the flow. There are no disturbances at the grid interface visible, the results at the grid transition are smooth. Figure 6.16 presents the mean velocity profile as  $u^+$  versus  $y^+$ . Comparison with the DNS data of [88] shows good agreement. The velocity profile is smooth at the grid transition, but a slight deviation in the region of grid transition is observable. Similar findings emerge from the turbulent fluctuations in Fig. 6.17,

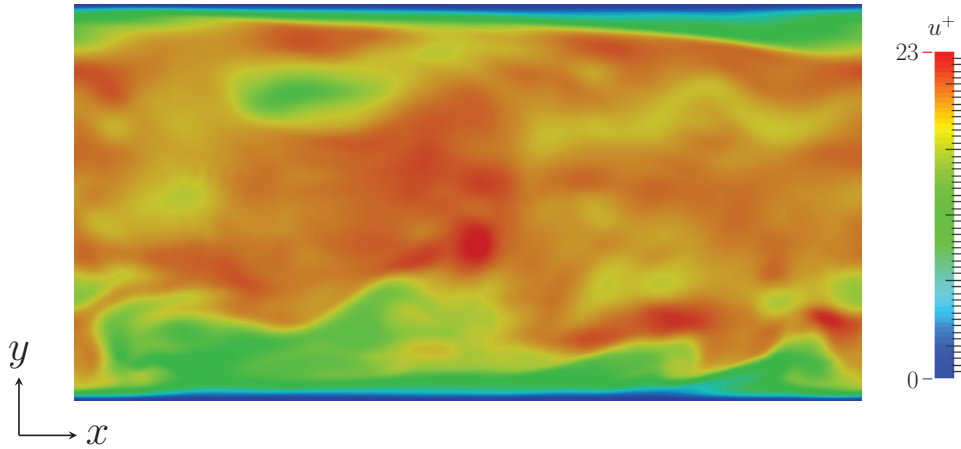


Figure 6.15: Instantaneous Velocity Field for Multi-Level Run

which additionally includes the grid structure as a direct reference. The results fit very well to the DNS and to the single-level case in the fine level remote from the interface. Differences between the SAM-Lattice solution and the reference and the single-level case, respectively, occur in the area near the grid interface. The change of grid resolution seems to introduce slight disturbances, which result in higher values of the turbulent fluctuations and are nonphysical. Thus the flow is more turbulent, i.e., has a higher turbulent intensity, what also induces the wider mean velocity profile in this region. Nevertheless, the results are feasible, since the fluctuations are very sensitive quantities.

The turbulent kinetic energy spectrum for the multi-level run is depicted in Fig. 6.18 at  $y^+ \approx 34$ . We get the same findings from the energy spectrum as in the single-level case. All three ranges in the spectrum are recognizable, in the inertial range the slope of  $-5/3$  is well reproduced. At a dimensionless wave number of  $\kappa'' \approx 75$  a small peak is detectable. It is assumed that this peak is also caused by the disturbances introduced from the grid transition.

Outcomes of the multi-level LES for turbulent channel flow prove that the scheme is feasible, but the accuracy in the region of the grid interface is slightly reduced. Thus, grid interfaces should be avoided in regions of interest as far as possible.

## 6.2.2 Wall Modeled Approach

After verification of the wall resolved approach, the wall modeled scheme will be verified in this subsection. For this purpose, again turbulent channel flow is used, but now at a friction Reynolds number  $Re_\tau = 2000$  what corresponds to a centerline Reynolds number  $Re_c \approx 50000$ . The fluid domain is still according to Fig. 6.6 with an equal case setup as for  $Re_\tau = 180$ . The simulation parameters, which differ to the calculations at  $Re_\tau = 180$  are specified in Table 6.13.

The first grid nodes for the test case are placed at  $y^+ = 50$  and the walls are placed in the middle of the first cell, i.e., halfway between wall and fluid node. Thus, the mesh size is fixed

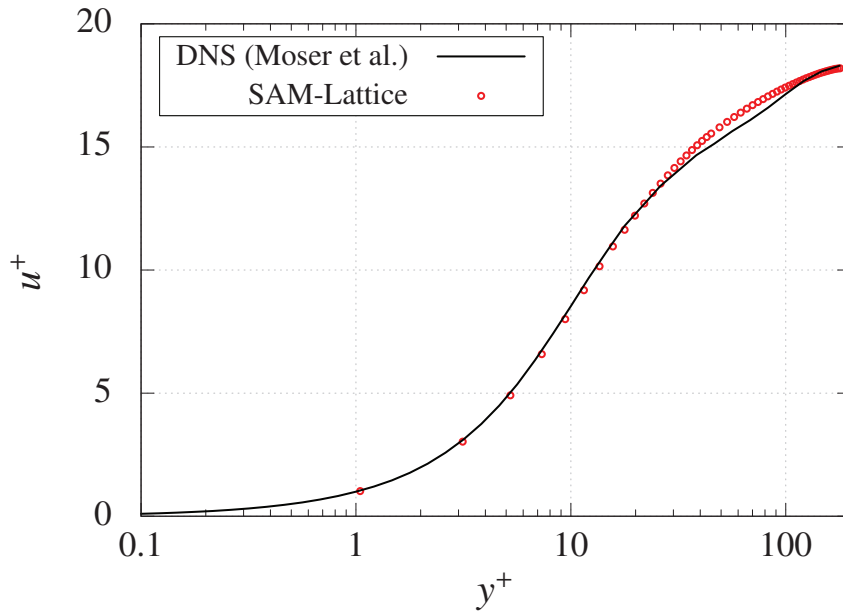


Figure 6.16: Mean Velocity Profile for Multi-Level Run

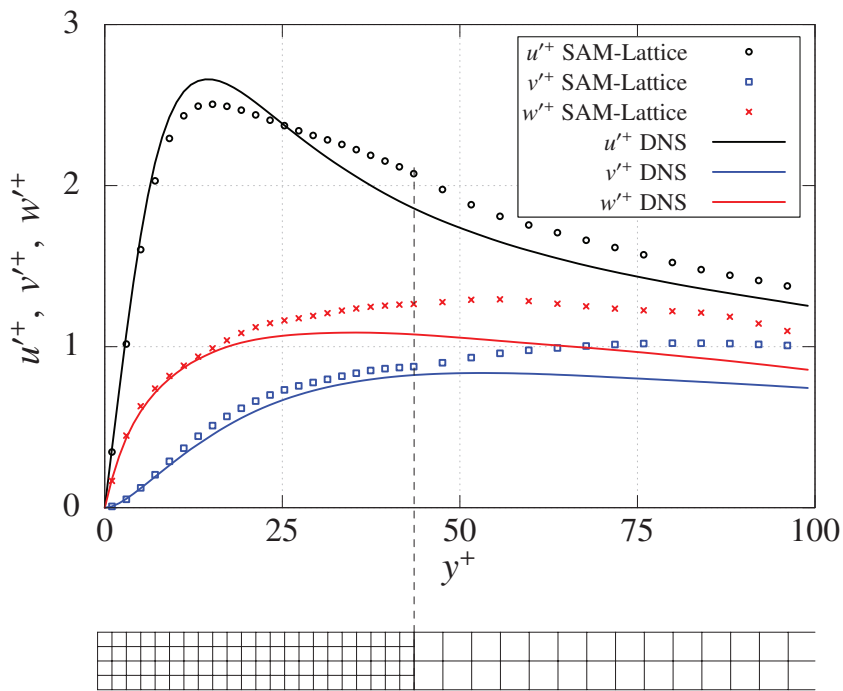


Figure 6.17: Turbulent Velocity Fluctuations for Multi-Level Run

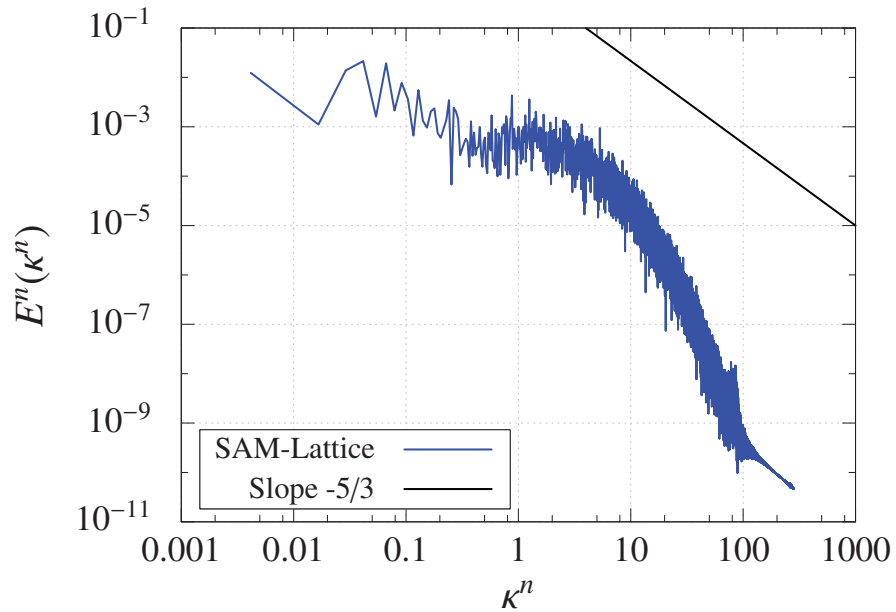


Figure 6.18: Turbulent Kinetic Energy Spectrum for Multi-Level Run

Table 6.13: Parameters for Turbulent Channel Flow at  $Re_\tau = 2000$ 

Parameter	Value	Unit
Friction Reynolds number $Re_\tau$	2000	—
Friction velocity $u_\tau$	0.0278	$m/s$
Acceleration $ a $	$7.728 \cdot 10^{-4}$	$m/s^2$
Cells per $H$	20	—
Spacing $\Delta x$	0.05	$m$
Time step $\Delta t$	$4.12 \cdot 10^{-3}$	$s$
Lattice Mach number $Ma$	0.1	—
Averaging period $\Delta T$	$60 \frac{H}{u_\tau}$	$s$

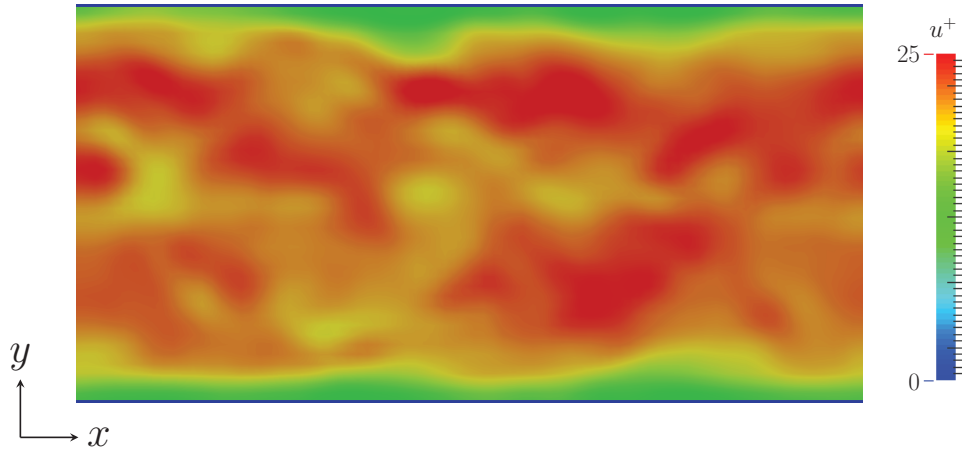


Figure 6.19: Instantaneous Velocity Field for Wall Modeled Approach ( $C_s = 0.1$ )

to be  $\Delta x^+ = 100$  in wall units. By means of the two layer approach, as introduced in Subsection 4.3.1, the turbulent flow is treated. At both channel walls the boundary condition is set by the wall function in combination with the shear anti bounce back scheme, due to stability reasons. The eddy viscosity calculation is performed for the wall adjacent fluid nodes by the mixing length model (Eq. (4.35)) and for all other nodes by the Smagorinsky model. As before, only the MRT collision model is utilized thanks to its higher stability. Initialization is performed with equilibrium at zero velocity, as used for the preceding simulations.

An instantaneous snapshot of the flow field by means of  $u^+$  in Fig. 6.19 depicts the turbulent nature of the flow. Figure 6.20 presents the mean velocity profile via  $u^+$  vs.  $y^+$  for different values of the Smagorinsky constant  $C_s$ . For all values of  $C_s$ , the velocity at the first node matches very good to the DNS solution. This confirms that the boundary condition and the wall function, respectively, are working correctly. Furthermore, we see that for the commonly used value of  $C_s = 0.1$ , the correct centerline line velocity of the DNS is met. But, the velocity profile in between the first node and the centerline deviates from the DNS. By increasing  $C_s$  this difference and the centerline velocity are rising, as the case  $C_s = 0.125$  reveals. For  $C_s = 0.075$ , the solution adapts to the DNS profile, which is mostly well reproduced. Only the centerline velocity is underestimated by approximately 4.7%. If we look at the turbulent fluctuations in Fig. 6.21 for  $C_s = 0.1$ , the origin for the wider velocity profile gets obvious. All velocity fluctuations are overpredicted towards the channel center, what is nonphysical and causes higher mean velocities. A precise reason for this behavior could not be found, yet. Among the possible factors is the grid spacing ( $\Delta x^+ = 100$ ), which is relative large for LES, see Section 4.5. An indication for this comes from the eddy viscosity ratio, which has a maximum value of approximately 36 and may be too high for this Reynolds number. Also spurious oscillations might be an influencing factor. The characteristic shape of  $u'$  is met quite well, for  $v'$  and  $w'$  the peaks near the channel wall are not reproduced.

Since the velocity profile is highly influenced by the Smagorinsky constant and the agree-

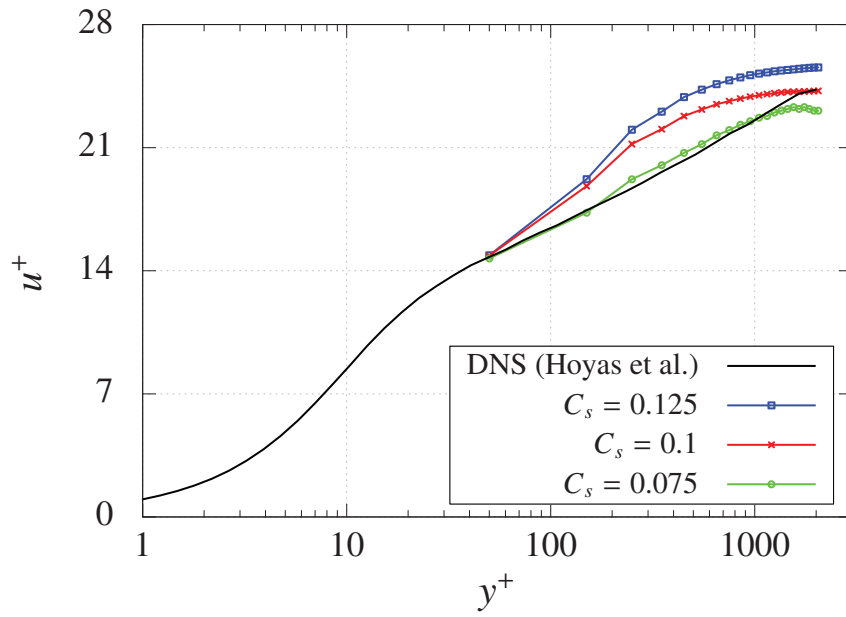


Figure 6.20: Mean Velocity Profiles for Wall Modeled Approach

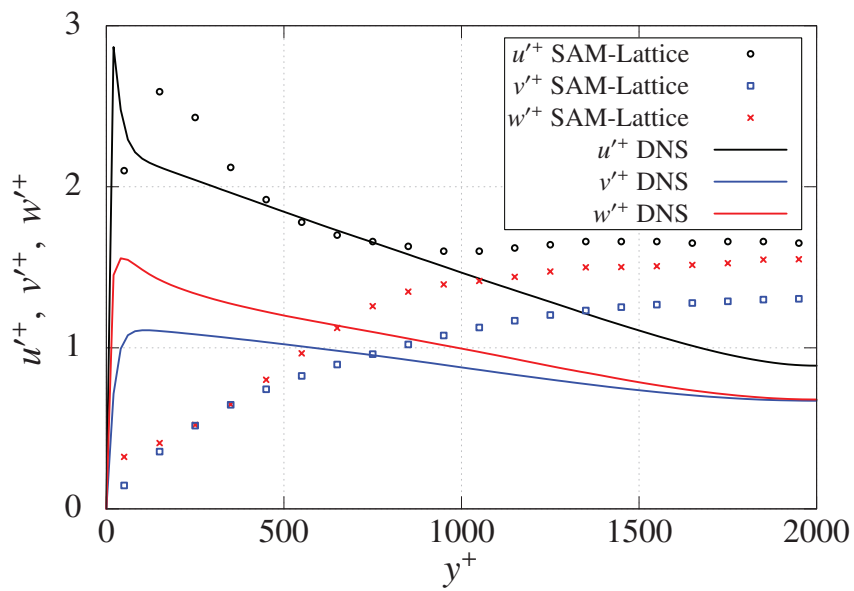


Figure 6.21: Turbulent Velocity Fluctuations for Wall Modeled Approach ( $C_s = 0.1$ )

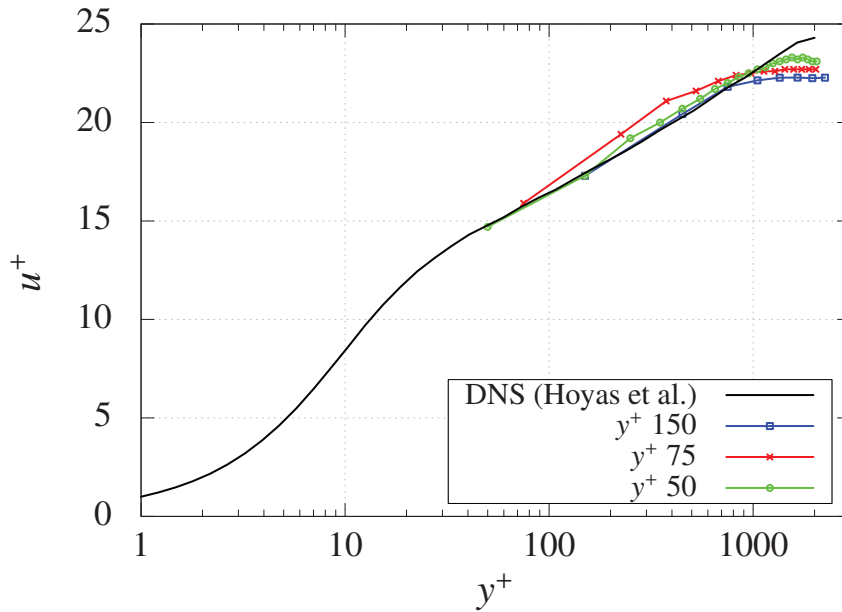


Figure 6.22: Mean Velocity Profiles for Different  $y^+$  ( $C_s = 0.075$ )

Table 6.14: Mesh Dependent Parameters for Turbulent Channel Flow at  $Re_\tau = 2000$

$y^+$	75	150
Cells per $H$	14	7
Spacing $\Delta x$ [m]	0.075	0.15
Time step $\Delta t$ [s]	$6.19 \cdot 10^{-3}$	$1.24 \cdot 10^{-2}$

ment to the DNS solution is much better for  $C_s = 0.075$ , it seems natural that more advanced eddy viscosity models can improve the solution. At best, a dynamic model increases or decreases  $C_s$  in the according regions, as it is required by the flow physics and thus adapts to the correct solution. The combination of a dynamic subgrid scale model and the wall modeled approach is suggested for future work.

The influence of the grid spacing and the wall distance of the first node, respectively, is investigated in Fig. 6.22 for the mean velocity profile. Additionally to the  $y^+ = 50$  grid, as specified in Table 6.13, two further grids with  $y^+ = 75$  and  $y^+ = 150$  are used. All parameters, which differ to the primary mesh are listed in Table 6.14. The Smagorinsky constant is chosen to  $C_s = 0.075$  for these three cases. From the results we see at first that for each mesh the first grid node lies on the DNS profile and verifies again the wall function implementation. Secondly, the solution approximates better the DNS for decreasing wall distance or spacing, respectively.



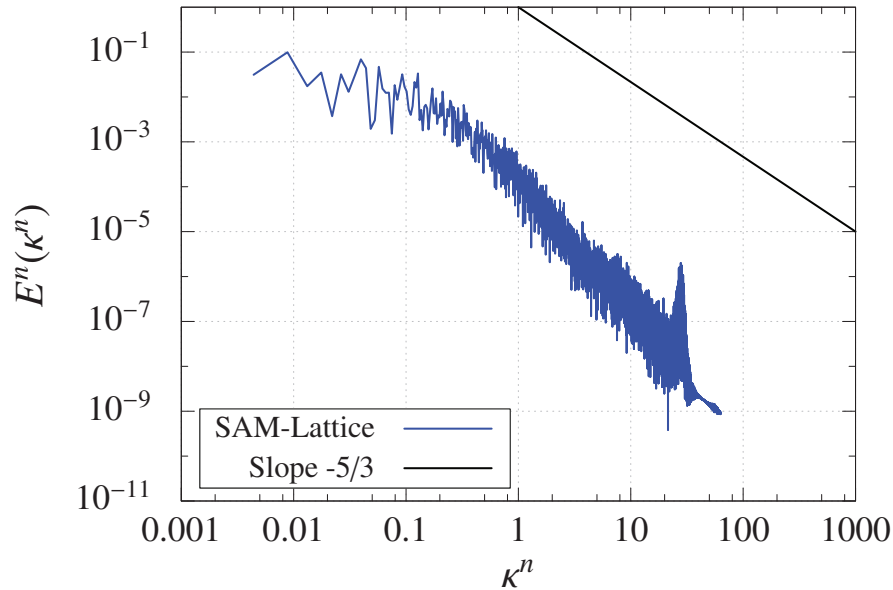


Figure 6.23: Turbulent Kinetic Energy Spectrum for Wall Modeled Approach ( $C_s = 0.075$ )

Figure 6.23 depicts the turbulent kinetic energy spectrum in the channel flow at a position  $y^+ = 450$  from the calculation with  $y^+ = 50$  mesh and  $C_s = 0.075$ . The inertial range is clearly reproduced and recognizable. Shortly before the cutoff length, a jump in  $E^n$  is visible, but the reasons therefor are not clear. Again, possibilities may be the large grid spacing or spurious oscillations, respectively.

The findings of this subsection reveal that the wall modeled approach captures well the mean velocity profile in turbulent channel flow. Furthermore a strong dependency of the results on the Smagorinsky constant is determined. The representation of the turbulent fluctuations is fair, but here room for improvement, e.g., by another subgrid scale model, is at hand. All in all, the correct implementation of the wall modeled approach is proven.

### 6.2.3 Synthetic Eddy Method

The remaining component of the proposed LES approach is the synthetic eddy method, used for generation of synthetic turbulence at the domain inlet. Verification for SEM is conducted for a decaying isotropic turbulence flow in an infinite domain. Figure 6.24 shows the domain, which is set periodically in  $y$  and  $z$ -direction. The test case is inspired by the experimental setup of [64]. At the inlet boundary of the domain a constant velocity  $u_0 = 12 \text{ m/s}$  in  $x$ -direction is specified, which is superimposed by turbulent fluctuations generated by SEM. The synthetic eddy method requires certain additional input parameters, which characterize the turbulence as explained in Subsection 4.3.2. Since we are assuming isotropic turbulence here, things are

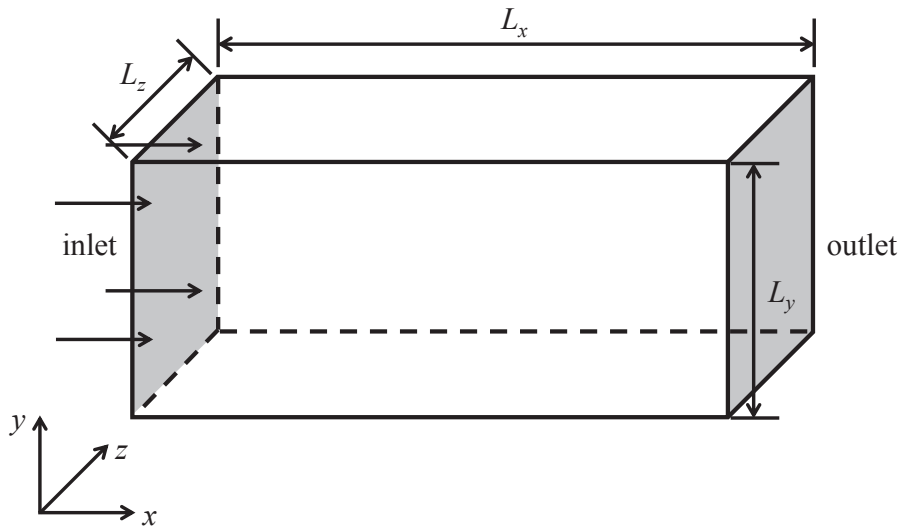


Figure 6.24: Domain for SEM Test Case

greatly simplified. It is adequate to specify the turbulent intensity  $I = 15.4\%$  and the turbulent length scale  $l = 0.25 \text{ m}$ , which are set equal to the values of the experiment [64]. The Dirichlet boundary condition with the resulting velocity at the inlet is performed with the Bouzidi scheme. At the outlet a constant gauge pressure is prescribed, which is numerically realized by an equilibrium condition. The distributions are again initialized with a zero velocity equilibrium and the MRT model is used. For convenience all simulation parameters are summed up in Table 6.15.

Since the focus of this test case is on the verification of SEM as a domain inlet condition, we will only consider the flow field at the domain inlet in the sequel. Figure 6.25 shows on the left hand side the instantaneous velocity components in the inlet plane by means of dimensionless velocity components for an arbitrary point in time. The sub-figures on the right hand side depict the dimensionless velocity components as a function of normalized time for a representative point in the middle of the inlet plane. For clarity, the average velocity is indicated in the graphs by the green line. All sub-figures of Fig. 6.25 illustrate the turbulent nature of the flow at the inlet. For a more detailed verification, the statistics of the flow in the inlet plane are calculated for the averaging period  $\Delta T$  and are combined in Table 6.16. The statistical results reveal that the turbulence is fairly isotropic and that the specified mean values are achieved. This data yields a turbulent intensity of  $15.46\%$ . Thus, SEM produces the desired turbulent flow at the domain inlet. The small differences between the resulting velocity statistics and the prescribed values are expected to get even smaller for extending the temporal average period.

On the basis of these results, verification of SEM is provided. To close this chapter, it can be concluded that the LES-LBM is now completely verified.

Table 6.15: Parameters for SEM Test Case

Parameter	Value	Unit
Domain size $L_x \cdot L_y \cdot L_z$	$8 \cdot 0.91 \cdot 1.22$	$m^3$
Periodic directions	$y, z$	–
Density $\rho$	1.225	$kg/m^3$
Kinematic viscosity $\nu$	$1.39 \cdot 10^{-5}$	$m^2/s$
Velocity at inlet $u_0$	12	$m/s$
Turbulent intensity at inlet $I$	$I = \frac{u'}{u} = 15.4$	%
Turbulent length scale at inlet $l$	$l = 0.25$	$m$
Gauge pressure at outlet $p_{out}$	0	$Pa$
Dimensionless velocity $u^n$	$u^n = \frac{u}{u_0}$	–
Dimensionless time $t^n$	$t^n = \frac{tL_z}{u_0}$	–
Spacing $\Delta x$	0.02	$m$
Time step $\Delta t$	$1.44 \cdot 10^{-4}$	$s$
Lattice Mach number $Ma$	0.15	–
Smagorinsky Constant $C_s$	0.1	–
Averaging period $\Delta T$	$15.63 \frac{L_x}{u_0}$	$s$

Table 6.16: Turbulent Statics at Inlet Plane

Component	Mean Velocity [m/s]	Velocity Fluctuation [m/s]
x-Velocity	12.005	1.866
y-Velocity	$-4.78 \cdot 10^{-2}$	1.854
z-Velocity	$3.72 \cdot 10^{-3}$	1.849

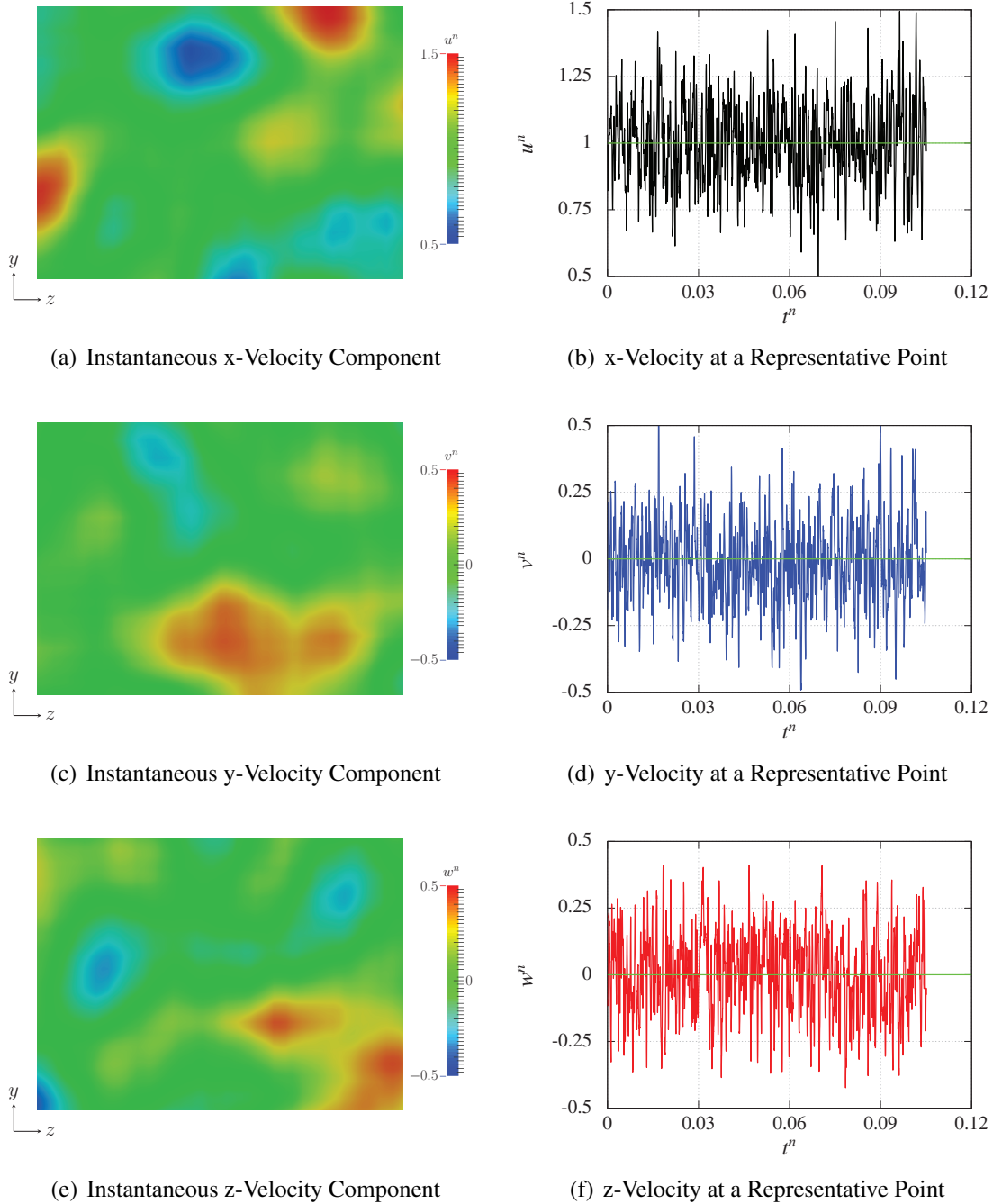


Figure 6.25: Results at Domain Inlet

---

### Pump Intake Flows - Complex Validation Cases

---

After verification of SAM-Lattice in the last section, the next step in the V&V process is validation. As already explained, validation is mandatorily based on comparisons with experimental data. In the scope of this section, validation of the LES module of SAM-Lattice will be performed for pump intake flows. Pump intake flows are complex flows, containing lots of coherent flow structures, and are of engineering interest due to their appearance in many plants. The Lattice Boltzmann Method exhibits many properties, which are beneficial for handling pump intake flows. In general, pump intake flows can be categorized into two groups: free surface and pressurized intake flows. For every group, a representative member is analyzed in the following, for which experimental data in literature is available. But, before coming to the actual test cases, an introduction to pump intake flows is given.

#### **7.1 Pump Intakes**

The withdrawal of water for closed piping systems from open water, e.g., reservoirs or rivers, is conducted with special hydraulic structures, which are called pump intakes or simply intakes. Such intakes can be found, for example, in industrial plants, power plants or waste water stations. In the simplest case, the intake structure consist of an approach channel, which leads the water to one or more pumps. Figure 7.1 depicts exemplary a free surface intake. For pressurized intakes there is an additional wall at the top of the structure and the intake is completely filled with water. Commonly, the intake structure includes flow guiding or flow controlling devices, like floor or sidewall splitters and cleaning devices. In combination with these additional devices the resulting geometrical structure of intakes is usually quite complicated. The flow inside pump intakes is in general very complex and contains commonly many coherent vortex

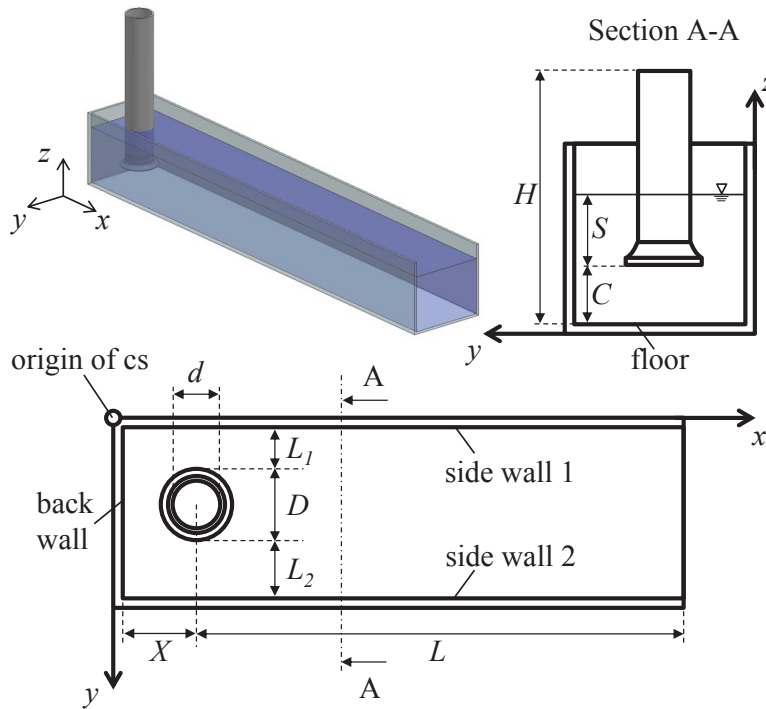


Figure 7.1: Exemplary Pump Intake and Geometrical Parameters

structures. If such vortices occur in the vicinity of the pump this can lead to unacceptable inflow conditions for the machine, which have consequences such as loss of efficiency, vibrations, or uneven impeller loading.

The vortices in intakes are classified, according to the location where they originate, as free surface or subsurface vortices. Hence, free surface vortices cannot occur in pressurized intakes. According to [55] free surface and subsurface vortices are classified into different types, depending on the vortex strength. The definitions of free surface vortex types are given in Table 7.1 and are schematically drawn in Fig. 7.2(a). Subsurface vortices can form from all wetted walls, i.e., the side walls, the back wall and from the floor in free surface intakes, cf. Fig. 7.1. Additionally, in pressurized intakes subsurface vortices can originate from the top wall. Classification of the different subsurface vortex types is listed in Table 7.2 and schematically illustrated in Fig. 7.2(b). According to state of the art criteria, there are only free surface vortices of type 1 or 2, respectively, and no subsurface vortices acceptable in intakes during operation [66].

Intake flow fields are observed to be highly unsteady and the vortices to be intermittent and meandering. The flow field depends on the geometrical parameters of the intake, as listed in Fig. 7.1, and the dimensionless parameters summarized in Table 7.3 [23]. Of course, the flow field is determined by the Reynolds number, which is build here with the mean velocity

Table 7.1: Types of Free Surface Vortices [66]

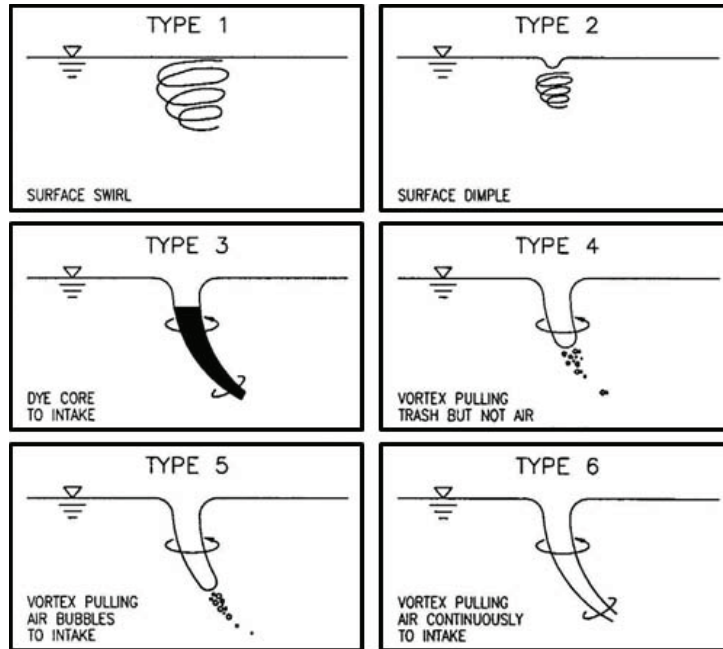
Type	Definition
1	Weak rotation of fluid at surface
2	Weak rotation with a dimple in the surface swirl
3	Dye core vortex - to be detected only by visualization
4	Vortex pulling trash from the surface downwards
5	Vortex pulling air bubbles from the surface, at times air bubbles entering the inlet nozzle
6	Constant air entraining vortex

Table 7.2: Types of Subsurface Vortices [66]

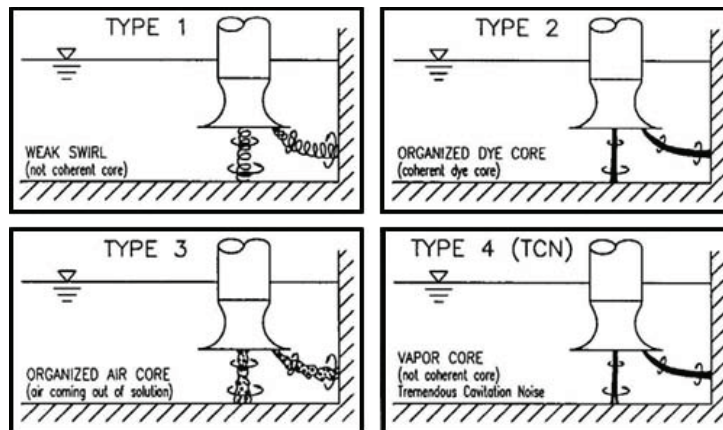
Type	Definition
1	Vortex visible with dye, but not coherent
2	Coherent swirl
3	Visible through air coming out of solution
4	Visible through a vapor core, coming out of the fluid

Table 7.3: Dimensionless Parameters

Parameter	Definition
Reynolds Number $Re$	$Re = \frac{u_d \cdot d}{\nu}$
Circulation Number $Nc$	$Nc = \frac{\Gamma \cdot d}{Q}$
Froude Number $Fr$	$Fr = \frac{u_{ch}}{\sqrt{g \cdot S}}$
Weber Number $We$	$We = \frac{u_d^2 \cdot \rho \cdot d}{\gamma}$



(a) Classification of Free Surface Vortices [66]



(b) Classification of Subsurface Vortices [66]

Figure 7.2: Vortices in Pump Intakes



$u_d$  and the diameter  $d$  in the pump. The circulation number specifies the circulation of the flow in the approach channel and can thus be interpreted as a specification of the inflow boundary condition. For normalization  $Nc$  uses the pump diameter  $d$  and the flow rate  $Q$ . Froude and Weber number are of importance for free surface intakes, since they express the behavior of the free surface. The Froude number, as ratio of inertial and gravitational forces, is build with the average velocity in the approach channel  $u_{ch}$ , the gravitational acceleration  $g$  and the submergence  $S$  of the pump. Lastly,  $We$  is the ratio between inertia and surface tension, which is build with  $u_d$ ,  $d$  and the surface tension  $\gamma$ . By all these geometrical and flow parameters, the flow can be influenced or changed.

Several design standards for pump intakes define rules and criteria to avoid the occurrence of vortices in the intake and to achieve acceptable operating conditions for the pumps; see, e.g., [5]. Usually, physical model investigations are conducted to evaluate and/or optimize the flow field in the intake. This model testing is expensive and time consuming, but as pump intakes are big and expensive constructions and the operability of the pumps is one of the most critical aspects for many plants, intensive design verification is needed. Model tests exhibit the disadvantage that it is not possible to hold all similarity rules at the same time [66] and thus engineering estimations are always necessary.

Due to these cost and modeling aspects, it is advisable to analyze the flow field in pump intakes by numerical methods. In this way, costs and turnaround times are reduced and by investigating the intake in real world dimensions, the need of estimating the similarity rules is omitted. Of course, concluding experimental investigations are mandatory to validate the calculations. In literature some studies regarding the numerical analysis of pump intakes can be found. The first numerical studies of pump intakes used steady RANS approaches with  $k - \epsilon$  turbulence models ([23, 117]). In [24] the influence of different turbulence models on the vortex structures in intakes was investigated for RANS schemes. Recent numerical analyses of intakes were performed using commercial solvers with steady and unsteady RANS approaches, e.g., [66, 91, 118], or using commercial or in-house LES solvers, e.g., [78, 122]. In all of the mentioned studies, the governing equations are based on a macroscopic description of the flow. Remembering the unsteadiness of the flow field in intakes, steady simulations are not feasible to capture accurately the dynamics of the flow.

Based on the flow characteristics and the geometrical properties of intakes we can directly highlight the advantages of the Lattice Boltzmann Method for dealing with pump intakes. LBM is perfectly suited for calculating flows with lots of geometrical details, as it can occur in intakes. The marginal preprocessing effort allows us to test various variants during the design process. As the nature of the flow and previous numerical studies in literature show, modeling of the flow in a transient manner is essential. Furthermore, the highly vortical structure of the flow is clearly better represented by a large eddy approach [122], what is obvious since less modeling and more direct description of the flow physics is used. Pump intake flows are fit and proper for LES, since the Reynolds number is typically  $\sim 10^5$ . LBM is very efficient in calculating transient flows, due to its intrinsic transient nature. In combination with the large

Table 7.4: Geometrical Parameters of Free Surface Intake

Parameter	Value	Parameter	Value	Parameter	Value
$d$	100 mm	$D/d$	1.5	$C/d$	1.0
$X/d$	1.1	$L_1/d$	0.65	$L_2/d$	0.85
$L/d$	25.86	$H/d$	6.0		

eddy approach proposed in this thesis, LBM appears as a high performance tool for intake and similar flow calculations.

The first computations of pump intakes by LBM have been reported by the author in [107, 108]. The ability of the proposed LES-LBM to capture the flow field in pump intakes is analyzed in the following sections. Special attention is turned to the potential of the numerical scheme to reproduce the transient vortex behavior of intake flows, which results in very complex flow structures and is challenging to model numerically. In the design process, the occurrence of vortices is the deciding design criteria for modification.

All calculation presented in this chapter rely on the MRT collision operator, due to its superior stability, and regular, non-refined meshes. The usage of multi-level grids in combination with the wall resolved LES approach is omitted here, since the stability criteria of the multi-level scheme ( $\Omega_0 < 1.995$ ) would require several hundreds of millions of cells and is thus not economical. Full resolution of boundary layers is not expected to be crucial for representing the vortices in intake flows, because the flow field is dominated by inertia effects (cf. [4]) and hence wall modeling should be appropriate.

## 7.2 Free Surface Intake

The benchmark case for the free surface intake is based upon the laboratory experiment of Okamura et al. [91]. Figure 7.1 depicts the geometrical structure of the intake, which consists of an approach channel and a single pump. All dimensions are chosen to replicate the experimental model for comparison with the measurements and are listed in Table 7.4. It is remarkable to notice that the pump is placed asymmetrically in the intake, what will provoke imbalances in the flow and promote vortices. This case is one of the best documented free surface intakes in literature, which contains all geometrical and flow parameters and simultaneously provides measurement data. We will use this simple structure, although it does not include flow guiding or cleaning devices and can thus not expose all capabilities of SAM-Lattice in handling complex geometries.

Consistently to the experiment, two different flow rates, i.e., discharges of the intake and two submergences are investigated. Out of these four parameters, three different operating points are constructed in [91] and considered for simulation. Table 7.5 lists the operating point

Table 7.5: Free Surface Intake - Operating Points (OP)

	<b>Flow rate <math>Q</math> [<math>\frac{m^3}{min}</math>]</b>	<b>Submergence <math>\frac{S}{d}</math></b>	<b>Re</b>	<b>We</b>	<b>Fr</b>
OP1	1.0	1.3	$2.122 \cdot 10^5$	6189.53	0.2139
OP2	0.6	1.3	$1.273 \cdot 10^5$	2228.23	0.1283
OP3	1.0	0.5	$2.122 \cdot 10^5$	6189.53	0.3448

dependent quantities and the according dimensionless parameters. The circulation number  $Nc$  is not specified, since the circulation is not provided in the experimental data.

Let us first explain the numerical model applied for the intake flow simulations. Since SAM-Lattice does currently not include multi-phase capabilities, an engineering workaround based on single phase flow is used. So, the air above the water is neglected and the free surface is assumed to be plane and treated as a frictionless wall. This modeling is motivated by the experimental findings and the dimensionless parameters. In the experiments the surface is described to be smooth, i.e., the amplitudes of the free surface waves are small. Due to the high Weber number  $We > 1000$ , the influence of the surface tension and the resulting shear stress at the free surface are negligible [23]. From an engineering point of view, this modeling is sufficient. Because, the most important result is if vortices are entering the pump, regardless if they are of single or multi-phase. Nevertheless, we have to assume that the dynamics of multi-phase, i.e., air entraining vortices are comparable to single phase vortices at the free surface. Analog to the experiment, the pump itself is modeled by a dummy. That means it is realized by a pipe with the shape of the pump casing, but without impeller, see [91] for details.

Numerically, the frictionless free surface is realized by the specular reflection bounce back scheme. The remaining solid walls of the intake and pump are modeled by the Mei scheme. At the outlet of the fluid domain, which is at the end of the pipe, see Fig. 7.1, a constant static pressure of 1 bar is specified by an equilibrium condition. The velocity for this equilibrium is extrapolated from the fluid domain by a zero gradient assumption. At the inlet of the fluid domain, which is the open end of the approach channel, the velocity according to the flow rate is specified. To generate a turbulent flow at the inlet, SEM is used and the fluctuations are superimposed to the mean velocity. Numerically, the velocity is prescribed by the Bouzidi scheme. Since no details about the turbulent statistics at the inlet are given in [91], isotropic turbulence with an intensity of 5% and a turbulent length scale of 7% of the channel inlet width is assumed. Of course, this assumption will introduce additional errors, if it is far apart from the experimental conditions. Smagorinsky's constant  $C_s$  is chosen to 0.1. The time step for the simulations is in the order of  $6 \cdot 10^{-5} s$ , what corresponds to a dimensionless time step of approximately  $1 \cdot 10^{-3} \cdot d/u_d$ . With this time step size the maximum lattice Mach number is about 0.1. Recording of transient results for instantaneous analyses and averaging is started after an initial period of  $300 \cdot d/u_d$  to obtain a statistically steady state solution. The recording of results

Table 7.6: Simulation Parameters for Free Surface Intake

Parameter	Value	Unit
Density $\rho$	1000	$kg/m^3$
Kinematic viscosity $\nu$	$1 \cdot 10^{-6}$	$m^2/s$
Gravitational acceleration $g$	9.81	$m/s^2$
Surface tension $\gamma$	$72.75 \cdot 10^{-3}$	$Pa$
Turbulent intensity at inlet $I$	5	%
Turbulent length scale at inlet $l$	$l = 0.07 \cdot (L_1 + D + L_2)$	$m$
Gauge pressure at outlet $p_{out}$	0	$Pa$
Characteristic length $d$	0.1	$m$
Characteristic velocity $u_d$	$u_d = \frac{Q}{(d/2)^2 \cdot \pi}$	$m/s$
Dimensionless coordinates $x^n, y^n$	$x^n = \frac{x}{d}, y^n = \frac{y}{d}$	–
Spacing $\Delta x$	0.003	$m$
Time step $\Delta t$	$\sim 6 \cdot 10^{-5}$	$s$
Lattice Mach number $Ma$	$Ma = \frac{1.5 \cdot u_d}{c_s} = 0.1$	–
Smagorinsky Constant $C_s$	0.1	–
Free MRT relaxation parameter $s_i$	1.0	–
Averaging period $\Delta T$	$300 \frac{d}{u_d}$	$s$

is performed for another period of  $300 \cdot d/u_d$ . Discretization of the fluid domain is done by a regular grid with a spacing of  $0.03 \cdot d$ , resulting in approximately  $7 \cdot 10^6$  cells in the domain for operating point 1 and 2, and approximately  $5 \cdot 10^6$  cells for operating point 3. For convenience all simulation parameters are summarized in Table 7.6.

The results of these simulations have already been published by the author in [107, 108]. In the following we will focus at the main vortex structures present in the intake at operating point 1 and 3 and compare them to measurements as far as these are provided by [91]. Results for operating point 2 and comparison with literature results of other numerical schemes can be found in the aforementioned publications of the author.

At first, we have to determine if the grid resolution is sufficient for LES. Since the experimental results do not provide measurements of the turbulent kinetic energy  $k$ , the ratio of eddy viscosity to molecular viscosity  $\nu_t/\nu$  is used, which is introduced as single-grid estimator in Section 4.5. The viscosity ratio shows values less than 20 in the intake itself and rises in the

pump dummy to values less than 50, what is acceptable since the results in the pump itself are not of particular importance.

Figures 7.3(a) - 7.3(d) illustrate the behavior of the free surface vortex structure at OP1 by means of streamlines, which are released closely below the free surface. The different snapshots show the vortex at four different instants of time. In Fig. 7.3(a) one vortex is clearly detected which arranges between the pump and side wall 1. A period later, no free surface vortex is present (Fig. 7.3(b)), only some swirl remains. The two remaining snapshots (Figs. 7.3(c) and 7.3(d)) reveal again one free surface vortex between the pump and side wall 1. The different snapshots demonstrate the change of the vortex location and its structure with time. In the experimental studies one air entraining free surface vortex, located between pump and side wall 1, is detected. This air entraining vortex is intermittent and present for approximately 70% of the measurement time (according to [91]). Furthermore, the free surface vortex is identified to meander. This shows a good qualitative agreement between experimental observations and simulation results for the free surface vortex at this operating point. Furthermore, it demonstrates the ability of the presented LES-LBM scheme to capture transient vortex behavior.

The dominating vortex structures in the intake at operating point 1 are illustrated in Fig. 7.3(e) by iso contours of the well-known Q-criterion (see e.g. [75]). In accordance to the previous streamline pictures, the core of the free surface vortex is detectable between the pump and side wall 1. Furthermore, different subsurface vortices can be identified. There are subsurface vortices from both side walls, from the floor, and from the back wall. A more detailed description of all vortex structures is given by streamlines of the time averaged in-plane velocity. For every boundary of the intake, these streamlines are plotted in a plane close to the physical boundary, what is depicted in Fig. 7.4.

In the free surface plane (Fig. 7.4(a)) the already known dominant vortex between the pump and side wall 1 can be seen. Additionally, another small vortex is visible in this plane right next to the dominant one. This small vortex is not documented in the experimental analysis. Maybe it is too weak for an experimental detection, compared to the dominant vortex. Validation of the free surface vortex position is performed by the measurements of [91] in Fig. 7.4(b). The multiple measurement locations in the experiment illustrate the meandering of the free surface vortex. As we can see, the time averaged position of the simulation fits excellently to the measured positions.

Unfortunately, there are no further measurements for the subsurface vortices provided at this operating point. So, we can only interpret the numerical results at the remaining planes. In the planes next to each side wall, Figs. 7.4(c) and 7.4(d), two vortices can be seen. The large vortices correlate to the vortex cores visualized by the Q-criterion. Additionally, there is one small vortex at ground level in each plane. It is noticeable that the vortices at side wall 2 look more distorted compared to side wall 1. The floor plane, Fig. 7.4(e) shows a floor attached vortex which is located under the pump bell (dashed line). In the back wall plane (Fig. 7.4(f)), there is one dominant vortex detectable, corresponding to the Q-criterion.

We will continue with operating point 3, now. Compared to OP1, the water level, i.e., the

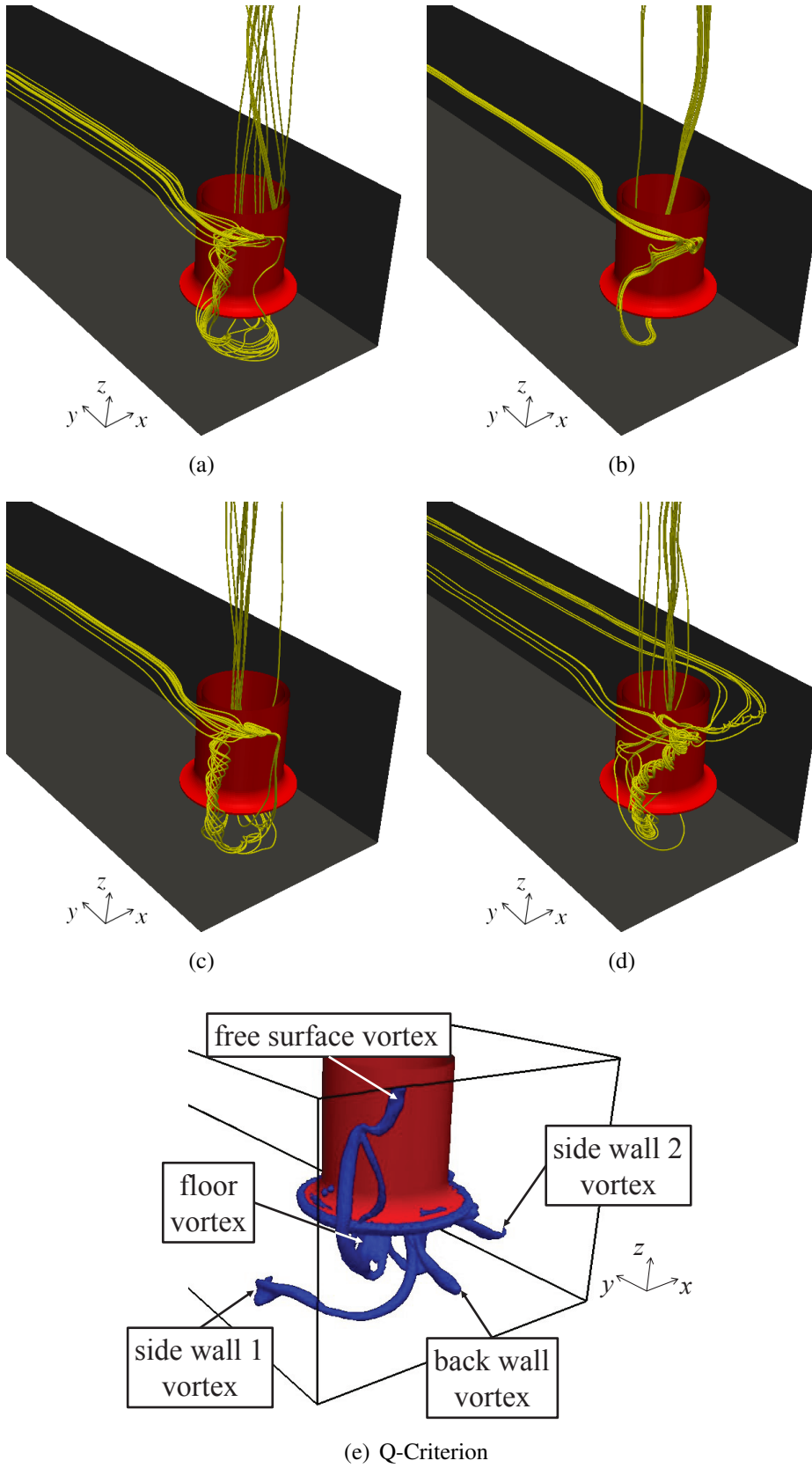
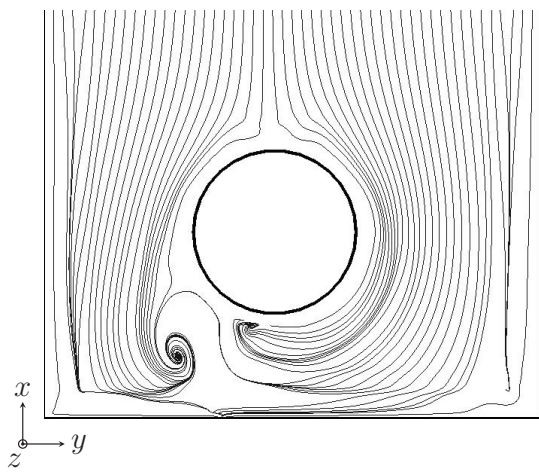
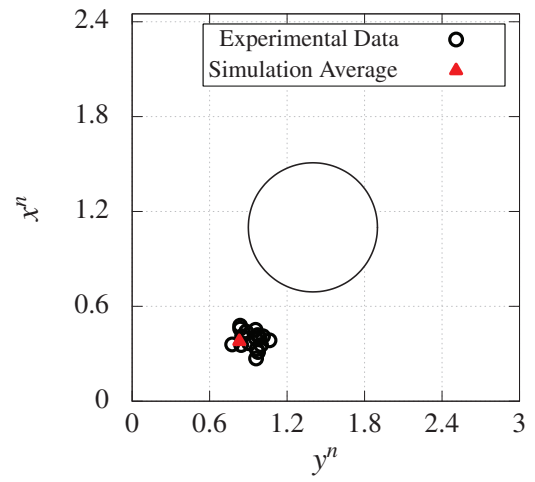


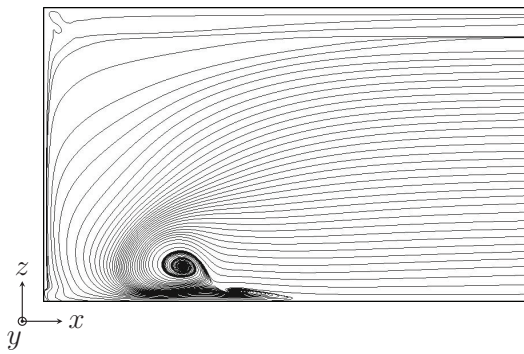
Figure 7.3: 3D Visualization of Vortex Structures at OP1



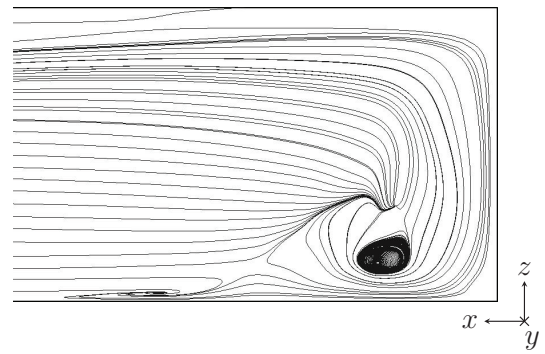
(a) Free Surface



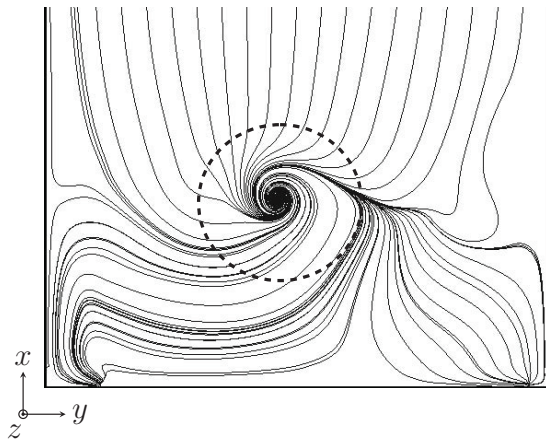
(b) Comparison with Experiment at Free Surface



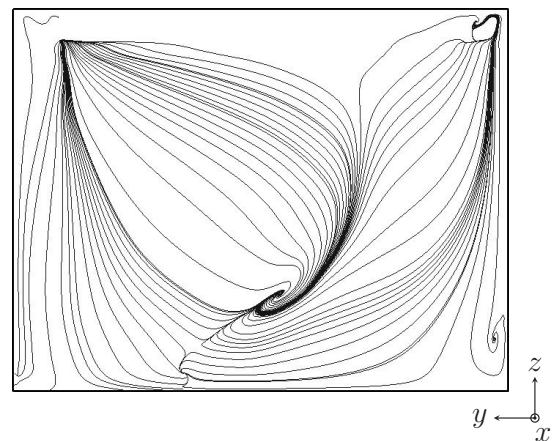
(c) Side Wall 1



(d) Side Wall 2



(e) Floor



(f) Back Wall

Figure 7.4: Time Averaged Surface Streamlines at OP1

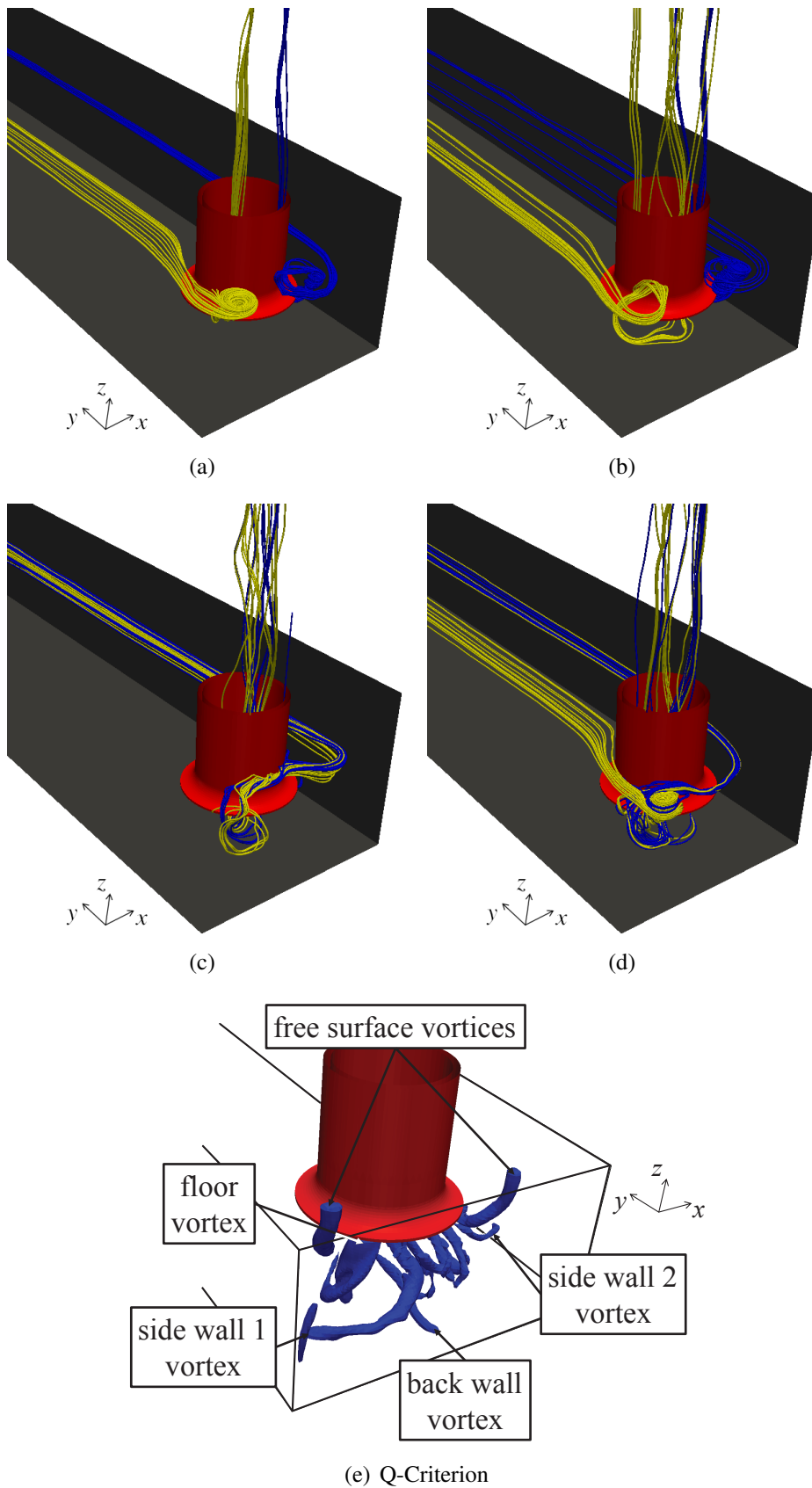


Figure 7.5: 3D Visualization of Vortex Structures at OP3



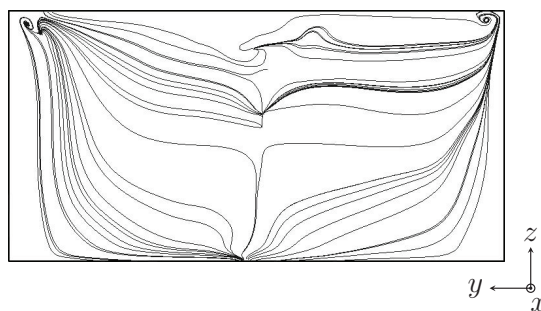
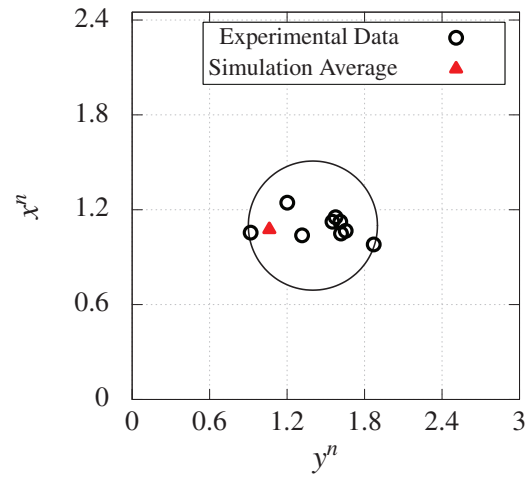
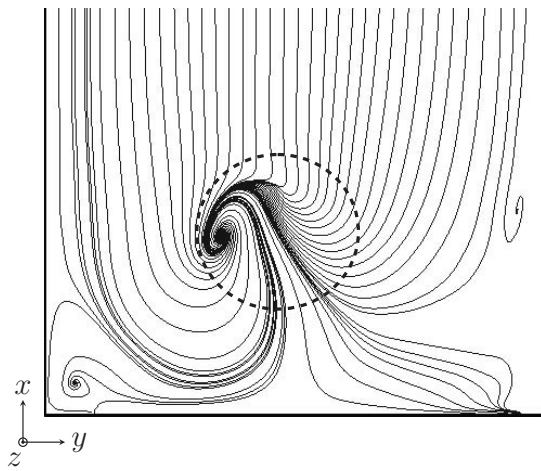
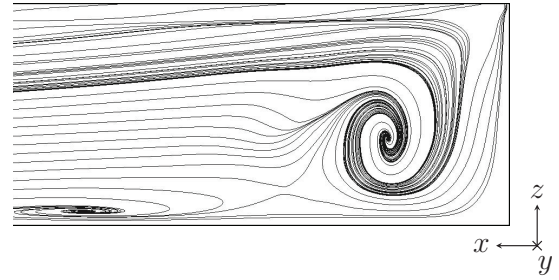
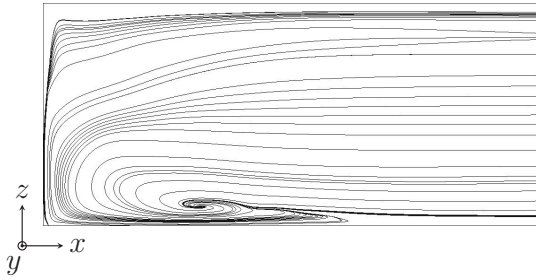
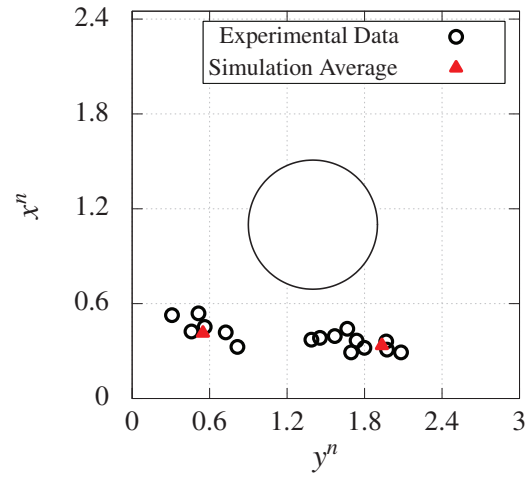
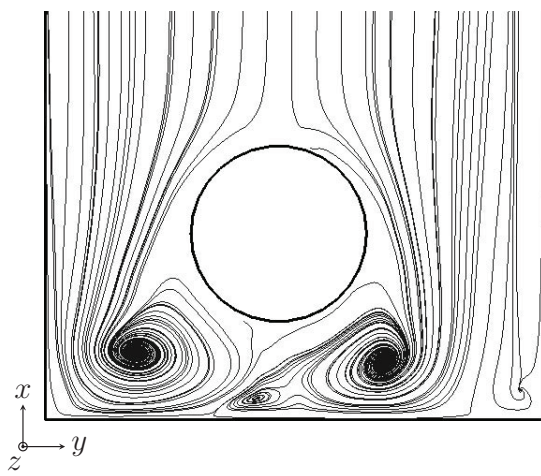


Figure 7.6: Time Averaged Surface Streamlines at OP3

submergence  $S/d$ , is reduced at a constant flow rate  $Q$ . As a result, the mean velocity in the approach channel  $u_{ch}$  is increased, what effects the vortices in the intake. The result presentation at OP3 is structured as before.

Figures 7.5(a) to 7.5(d) show the vortices at the free surface at four different instants of time. The flow topology at the free surface is different to OP1. In Fig. 7.5(a) two vortices behind the pump are present. For a better differentiation, the one next side wall 1 is again colored in yellow and the other one, next to side wall 2, in blue. A time period later, Fig. 7.5(b), the yellow vortex disappeared and left only some swirl, whereas the blue vortex is still traceable. Another period later, both free surface vortices are gone, see Fig. 7.5(c). It is remarkable that all streamlines run between the pump and side wall 2 here. In the last snapshot, Fig. 7.5(d), only the vortex at side wall 1 is regenerated and involves the flow passing between the pump and side wall 2, too. The findings for the free surface are again in good agreement with the experimental observation of [91], who reported two air entraining vortices, which are intermitten and meandering.

Figure 7.5(e) displays the dominating vortex structures at OP3 with the help of the Q-criterion. Beside the two free surface vortices, we see again one subsurface vortex originating from each intake boundary. A detailed analysis of these structures is given by means of streamlines of the time-averaged in-plane velocity, as done for OP1, in Fig. 7.6.

At the free surface, the two already identified vortices behind the pump can be detected in temporal average (see Fig. 7.6(a)). If we compare their position to the experimental locations in Fig. 7.6(b), we get a perfect agreement. A cavitating subsurface vortex (type 4) underneath the pump is reported in experimental investigations at this operating point. Of course, we can not reproduce cavitation with the single-phase model used in this simulation. But, the floor vortex is reproduced as shown by Fig. 7.6(e) and the time-averaged position of the floor vortex is reasonable in the area of measured instantaneous locations for this vortex, see Fig. 7.6(f). For the remaining boundaries, no measurement data is provided, so we can again only interpret the simulation results. The time-averaged solution shows one dominant subsurface vortex on each side wall plane. Analog to OP1, the one at side wall 2 is more distinctive. At the back wall (Fig. 7.6(g)), no dominant subsurface vortex can be identified in temporal average.

From these results for the free surface intake we can conclude that the LES-LBM scheme is feasible to reproduce the multiple vortices in the intake occurring at different operating points. Furthermore, transient features of the vortices, like meandering and intermittency are reproduced. Quantitative comparisons of the calculated and measured vortex positions show very good agreement. Thus, we can highlight that the LES-LBM is able to reproduce the complex flow and associated vortices in this pump intake and that the modeling assumptions applied to the free surface are adequate. More detailed quantitative analyses are given in the next section, since superior experimental results are available for that benchmark case.

Table 7.7: Geometrical Parameters of Pressurized Intake

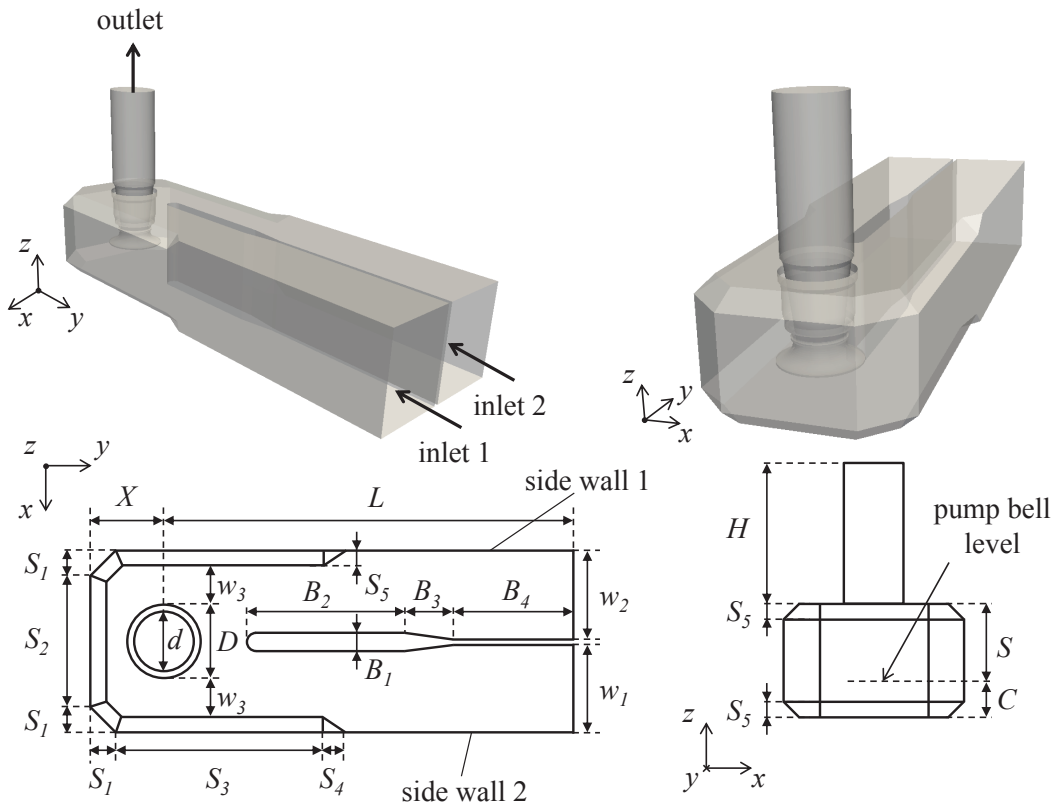
Parameter	Value	Parameter	Value	Parameter	Value
$D$	160 mm	$d/D$	0.8125	$C/D$	0.5
$X/D$	1.0	$L/D$	15.625	$S/D$	1.05
$w_1/D = w_2/D$	0.7125	$w_3/D$	0.53125	$H/D$	1.9
$S_1/D$	0.5	$S_2/D$	1.5	$S_3/D$	3.1875
$S_4/D$	0.3125	$S_5/D$	0.21875	$B_1/D$	0.25
$B_2/D$	2.15625	$B_3/D$	0.65625	$B_4/D$	11.6875

### 7.3 Pressurized Intake

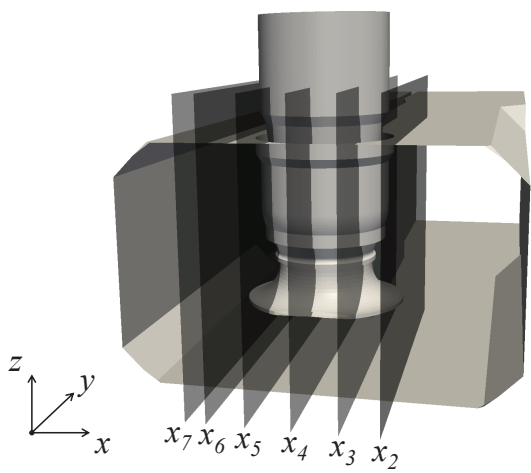
For validation of pressurized pump intake flow, the laboratory experiment published by [133, 134] is calculated. The geometry of the intake is depicted in Fig. 7.7(a) and the according dimensions are given in Table 7.7. This intake structure comes closer to a realistic pump intake, it contains one pump and several splitters in the corners of the pump sump. An additional baffle is used to create two separate approach channels. In the experimental studies, two dimensional PIV measurements are conducted in certain planes, which are each perpendicular to one of the principal axis and depicted in Figs. 7.7(b) and 7.7(c). The exact position of all used planes is given in Table 7.8, where the notation is according to the original source [134]. From these measurements, streamline plots and contour plots of the in-plane velocity and turbulent kinetic energy are available. Additionally, the intake has been studied numerically by Navier-Stokes based schemes (see, e.g., [122]).

In the benchmark case considered in the following, there is a large imbalance of the flow rate in approach channel 1,  $Q_1 = 0.905 \text{ m}^3/\text{min}$ , and channel 2,  $Q_2 = 0.385 \text{ m}^3/\text{min}$ , what induces an asymmetric flow in the pump sump and thus promotes the generation of vortices. Since the intake is completely filled with water, a single phase modeling is fully correct, if we exclude cavitation phenomena, and no additional modeling assumptions are needed. According to the experimental studies, the pump itself is again modeled by a dummy, i.e., it is realized by a pipe with the shape of the pump casing but without impeller, see [134] for details.

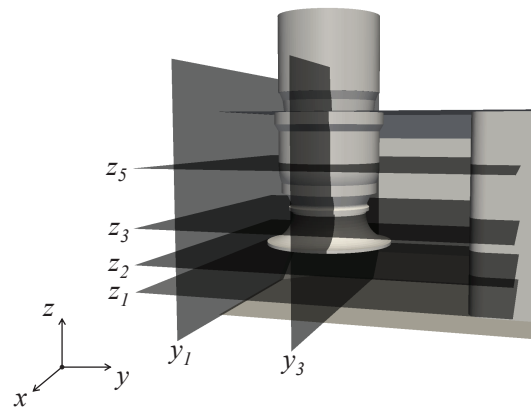
Setting up the numerical model for the pressurized intake is comparable to the free surface intake. The outlet of the fluid domain, which is at the end of the pipe, see Fig. 7.7(a), is represented by a pressure condition. A constant static pressure of 1 bar is specified by an equilibrium condition. The velocity for this equilibrium is extrapolated from the fluid domain by a zero gradient assumption. At the inlets of the fluid domain, which are the open ends of the associated approach channels, the velocities according to flow rates are specified. For generation of turbulent fluctuations at the inlet, SEM is used and the fluctuations are superimposed to the



(a) Intake Structure and Geometrical Parameters



(b) Measurement Planes  $x$



(c) Measurement Planes  $y$  and  $z$

Figure 7.7: Pressurized Pump Intake

Table 7.8: Evaluation Planes

Name	Position	Name	Position	Name	Position
$x_2$	$x/D = 1.75$	$x_3$	$x/D = 1.51875$	$x_4$	$x/D = 1.25$
$x_5$	$x/D = 0.98125$	$x_6$	$x/D = 0.75$	$x_7$	$x/D = 0.625$
$y_1$	$y/D = 0.3125$	$y_3$	$y/D = 1.125$	$z_1$	$z/D = 0.3125$
$z_2$	$z/D = 0.5$	$z_3$	$z/D = 0.75$	$z_5$	$z/D = 1.15625$

mean velocity. Numerically, the velocity is prescribed by the Bouzidi scheme. Again no details about the turbulent statistics at the inlets are given in the documentation of the experimental investigations, so isotropic turbulence with an intensity of 5% and a turbulent length scale of 7% of the width of the approach channels is assumed. All solid walls are modeled by the wall function in combination with the shear anti bounce back scheme. The Smagorinsky constant  $C_s$  is set to 0.1, as usual, and the time step for the simulations is in the order of  $2.5 \cdot 10^5 s$ . Resulting from the time step size, the maximum lattice Mach number is approximately 0.1. After an initial period of  $100 \cdot d/u_d$  a statistically steady state is reached and after  $150 \cdot d/u_d$ , the recording of transient results for averaging and instantaneous analyses is started. The recording procedure ranges over another period of  $150 \cdot d/u_d$ , what corresponds to approximately 12 s of real time. Discretization of the fluid domain is done by an regular grid with a spacing of  $0.0125 \cdot D$ , what produces approximately 32 million cells. The simulation is run on 256 CPUs within 30 hours of wall clock time. For convenience all simulation parameters are summarized in Table 7.9.

Let us start the result analysis with the evaluation of the grid resolution by means of single grid estimators. The viscosity ratio  $\nu/\nu_t$  shows values less than 15 in the intake itself and rises in the pump dummy up to values less than 37.5, what is acceptable. For a further estimation, Fig. 7.8 depicts the turbulent kinetic energy spectrum at a point  $P = (x^n = 1.25, y^n = 1, z^n = 0.35)$  directly under the pump center. We see that the inertial range, indicated by the  $-5/3$  slope, is resolved at this point. Both single grid estimators indicate the suitability of the grid. For a general overview of the vortex structures present in the intake, Fig. 7.9 visualizes the vortex cores in temporal average by means of the  $Q$ -criterion. Four large vortices can be identified at first glance: one from each side wall, one from the floor and one from the top wall. The side wall 1 vortex is weaker than the remaining vortices, what is indicated by a lower value of  $Q$  and the blue color, respectively.

To start the validation procedure, we will compare the time averaged in-plane streamlines of the numerical to the experimental results in Figs. 7.10 to 7.12 for the introduced measurement planes. The numerical results are always drawn on the left hand side of the figures. The experimental results are the shown on the right hand side of the according figures. Due to technical reasons of the PIV system, experimental results are only available for a certain area of the measurements planes. This area is indicated by a red box in the numerical plots. Furthermore,

Table 7.9: Simulation Parameters for Pressurized Intake

Parameter	Value	Unit
Density $\rho$	1000	$kg/m^3$
Kinematic viscosity $\nu$	$1 \cdot 10^{-6}$	$m^2/s$
Gravitational acceleration $g$	9.81	$m/s^2$
Flow rate at inlet 1 $Q_1$	0.905	$m^3/min$
Flow rate at inlet 2 $Q_2$	0.385	$m^3/min$
Characteristic velocity $u_d$	$u_d = \frac{Q_1+Q_2}{(d/2)^2 \cdot \pi}$	$m/s$
Reynolds number $Re$	$Re = \frac{u_d \cdot d}{\nu} = 2.106 \cdot 10^5$	–
Dimensionless velocity $u^n$	$u^n = \frac{u}{u_d}$	–
Dimensionless turbulent kinetic energy $k^n$	$k^n = \frac{k}{u_d^2}$	–
Dimensionless vorticity $\omega^n$	$\omega^n = \frac{\omega \cdot d}{u_d}$	–
Dimensionless coordinates $x^n, y^n, z^n$	$x^n = \frac{x}{D}, y^n = \frac{y}{D}, z^n = \frac{z}{D}$	–
Turbulent intensity at inlet $I$	5	%
Turbulent length scale at inlet $l$	$l = 0.07 \cdot w_1$	$m$
Gauge pressure at outlet $p_{out}$	0	$Pa$
Spacing $\Delta x$	0.002	$m$
Time step $\Delta t$	$\approx 2.5 \cdot 10^{-5}$	$s$
Lattice Mach number $Ma$	$Ma = \frac{3 \cdot u_d}{c_s} = 0.1$	–
Smagorinsky Constant $C_s$	0.1	–
Free MRT relaxation parameter $s_i$	1.0	–
Averaging period $\Delta T$	$150 \frac{d}{u_d}$	$s$

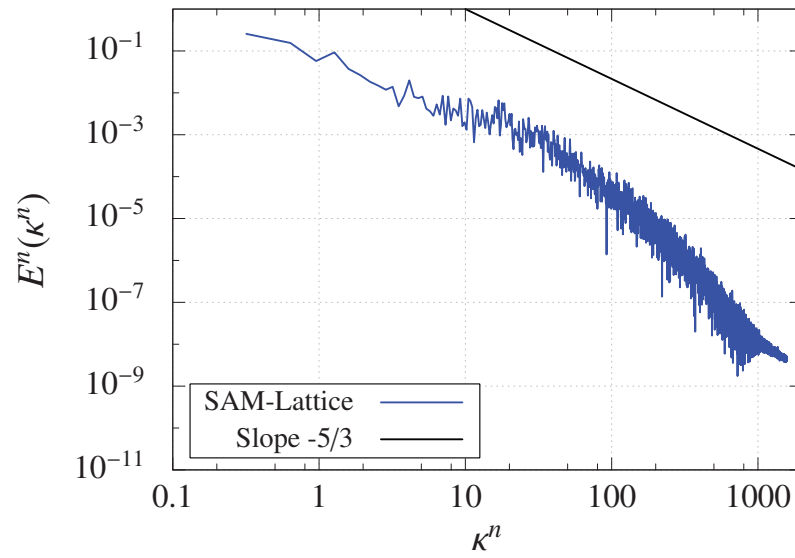


Figure 7.8: Turbulent Kinetic Energy Spectrum under Pump Center

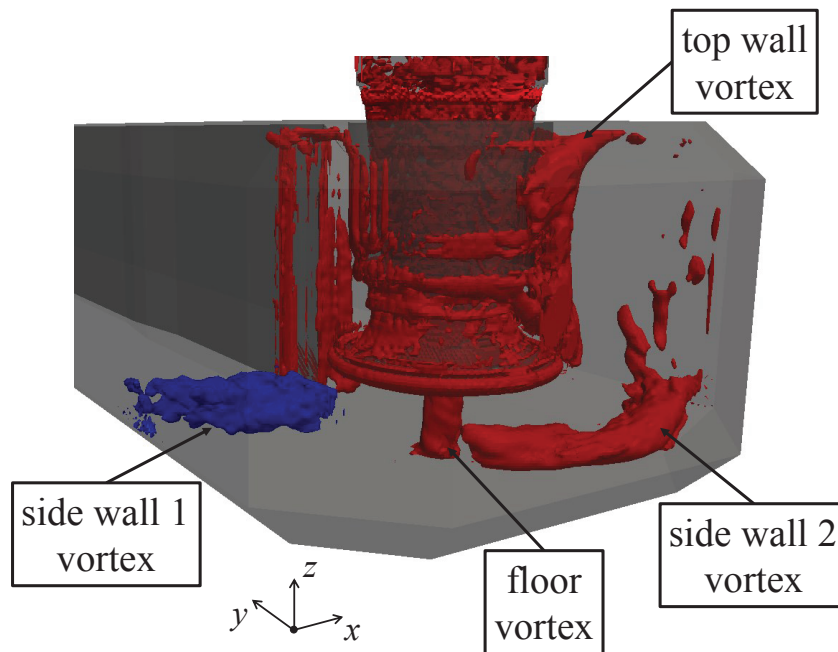


Figure 7.9: Time Averaged  $Q$ -Criterion

no results are provided in the pump dummy itself. Additionally to the measurement window, the pump bell level and the shape of the pump bell are denoted by red lines in the numerical outcomes.

Figures 7.10(a) and 7.10(b) face numerical and experimental results for the  $x_2$  plane. The numerical result reveals two vortices in temporal average, which can be associated to the top and side wall 2 vortex in Fig. 7.9. They are both outside of the measurement window. In accordance to the numerical findings, the experimental result does not show a vortex in the measuring zone. But, the converging streamlines in the experimental picture are in agreement to numerics and suggest a similar vortex towards the bottom. In the  $x_3$  plane, the side wall 2 and the top vortex can once again be identified; see Fig. 7.10(c). The side wall 2 vortex is now located in the measuring zone, but cannot be found in the PIV results in Fig. 7.10(d). This may indicate that the side wall vortices in the experiment is flatter and/or shorter. The remaining flow structure in the experimental window is matched well by the simulations. Next, we look at the  $x_4$  plane in Fig. 7.10(e), which cuts through the center of the pump. In addition to the top wall vortex behind the pump, which is getting bigger from  $x_2$  to  $x_4$ , another top wall vortex in front of the pump is predicted. Conformity with the PIV result, Fig. 7.10(e), is found for the streamline behavior in the measurement area.

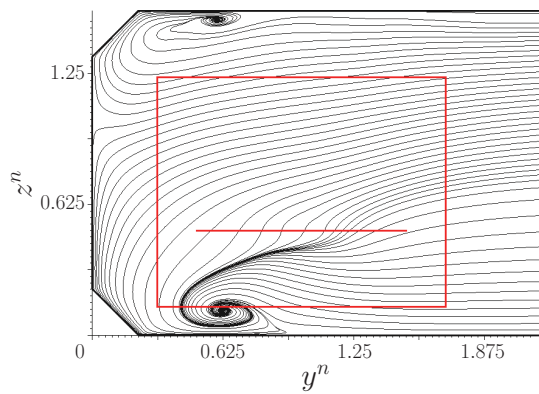
If we go on to the next plane  $x_5$ , which is shown in Fig. 7.11(a), we can only identify one vortex at the top wall in front of the pump. The streamlines in the experimental area agree well with the experiment, Fig. 7.11(b). Figure 7.11(c) examines the  $x_7$  plane. Here, the side wall 1 vortex is clearly visible at the right end of the measurement window. In the outcome of PIV (Fig. 7.11(d)), this vortex appears as well. The  $y$  position of the vortex in simulation and experiment match very well, whereas the  $z$  position is underestimated in the numerics.

For the  $y$  planes no experimental streamline pictures are available in literature. To gain insight in the flow pattern, Fig. 7.11(e) illustrates the numerical results in the  $y_3$  plane. Here, one small vortex at the top wall on the left side of the pump is present.

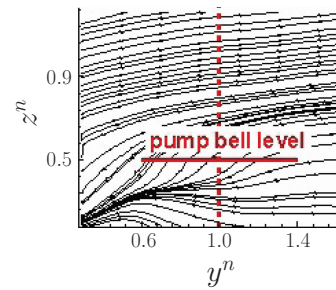
The flow structure for the  $z$  planes is given in Fig. 7.12. In the  $z_1$  plane, the floor vortex is the dominating structure; see Fig. 7.12(a). Besides this, two small vortices at the corners of the back wall splitters are detectable. In the PIV result in Fig. 7.12(b), the floor vortex is outstanding, too. The numerical and experimental positions of the floor vortex match excellently. For the  $z_3$  planes things are a bit different. The vortex of the experiment (Fig. 7.12(d)) at the bottom of the PIV window is reproduced in the simulation. However, an additional vortex in front of the pump is forecast by the calculation, which is not indicated in the experiment. A similar flow pattern is found in the  $z_5$  plane. Numerics predict three vortices in total, a small one in front of the pump, another small one at side wall 1 and a large one right below the PIV window. The large vortex is present in the experiment, its  $y$  position shows very good agreement between calculation and experiment, whereas the  $x$  positions differ.

For more quantitative comparisons, the literature on the experimental studies provides contour plots of the absolute in-plane velocity for some of the measurement planes. The contour plots shown in the following are based on the non-dimensional velocity  $u^n$ . In the  $x_6$  plane,

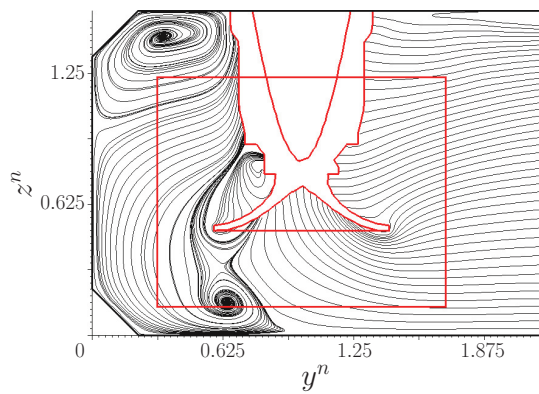




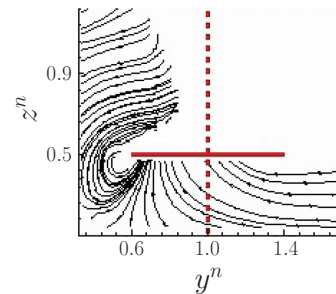
(a) Plane  $x_2$



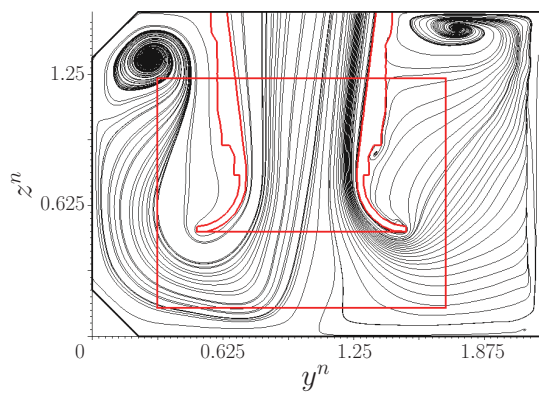
(b) Plane  $x_2$  (Experiment) [123]



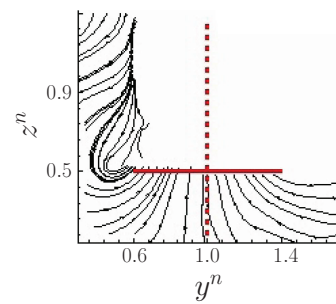
(c) Plane  $x_3$



(d) Plane  $x_3$  (Experiment) [123]



(e) Plane  $x_4$



(f) Plane  $x_4$  (Experiment) [123]

Figure 7.10: Time Averaged In-Plane Streamlines [Reprint of figures (b), (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.]

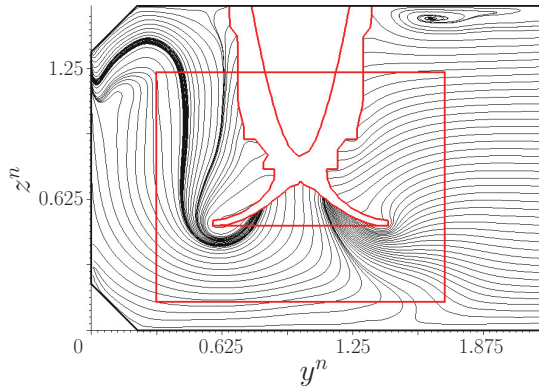
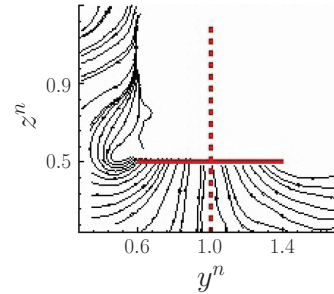
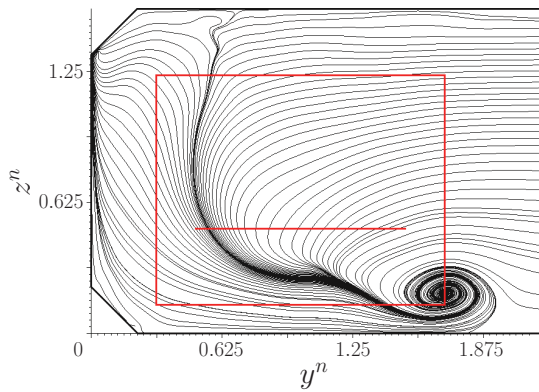
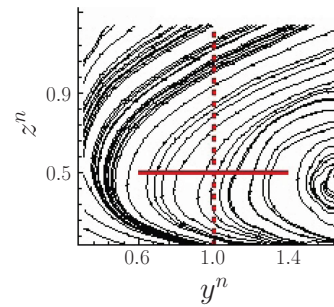
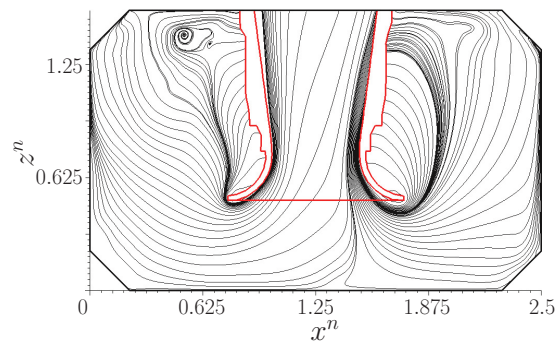
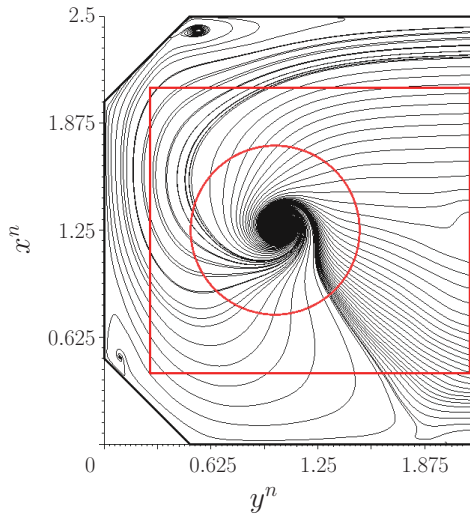
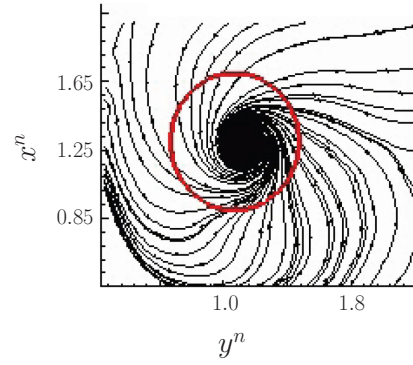
(a) Plane  $x_5$ (b) Plane  $x_5$  (Experiment) [123](c) Plane  $x_7$ (d) Plane  $x_7$  (Experiment) [123](e) Plane  $y_3$ 

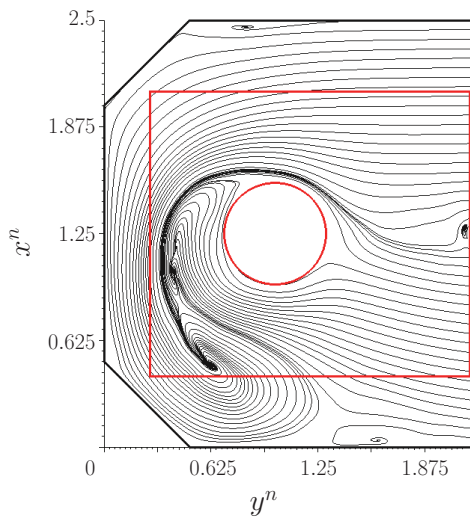
Figure 7.11: Time Averaged In-Plane Streamlines (2) [Reprint of figures (b) and (d) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.]



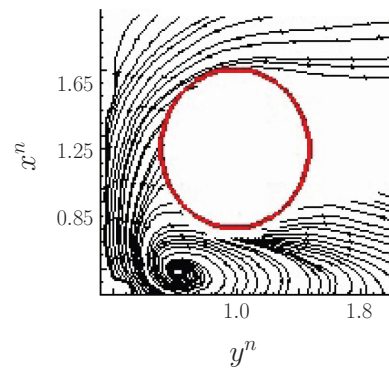
(a) Plane  $z_1$



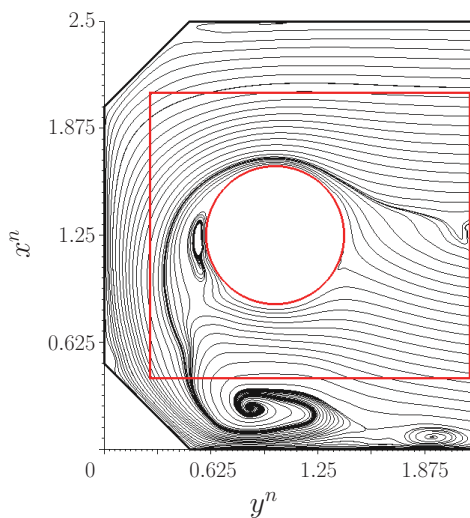
(b) Plane  $z_1$  (Experiment) [123]



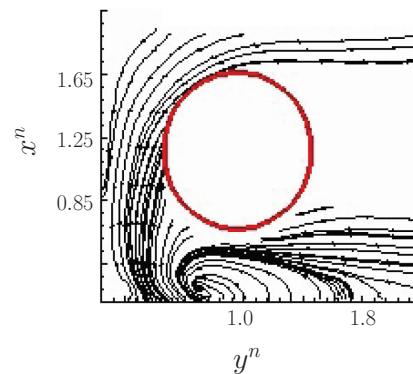
(c) Plane  $z_3$



(d) Plane  $z_3$  (Experiment) [123]



(e) Plane  $z_5$



(f) Plane  $z_5$  (Experiment) [123]

Figure 7.12: Time Averaged In-Plane Streamlines (3) [Reprint of figures (b), (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.]

Fig. 7.13(a), LES-LBM predicts the velocity quite well, all distinctive features of the PIV result in Fig. 7.13(b) are reproduced. For the  $x_7$  plane the conformity between calculation and measurement is fair. In the simulation, the high velocity area above the pump bell level is not as much developed as in the experiment, see Figs. 7.13(c) and 7.13(d). Good agreement is found between Figs. 7.13(e) and 7.13(f) for the  $y_1$  plane. The numerical outcomes contain the characteristics of the PIV image.

Equal conclusions hold for the  $y_3$  plane, depicted in Figs. 7.14(a) and 7.14(b). Numerics represent the measurements excellent in this section. (Remember that no PIV results inside the pump are available.) The two last pictures show  $z$  planes. In Figs. 7.14(c) and 7.14(d) the findings for the  $z_1$  plane are displayed. Both results agree quite well, the simulation manifests a slightly bigger area of high velocity around the center of the floor vortex. In the  $z_2$  plane, which is located directly under the pump bell (Figs. 7.14(e) and 7.14(f)), the simulation predicts high velocities nearly over the total bell mouth. In contrast, the PIV data reveals only a high velocity area around the center of the floor vortex or pump, respectively.

In addition to the contour plots for the mean velocity, the literature contains plots for the in-plane turbulent kinetic energy, which is build from the fluctuations of the in-plane velocity components. Again, we will compare the outcomes of PIV and numerics in different planes, for which experimental plots are available. As always, results are shown dimensionless by means of  $k^n$ . From these data, analyses of the reproduction of turbulent statistics can be drawn. We will start with the  $x_4$  plane, which cuts through the pump center, in Figs. 7.15(a) and 7.15(b). In the floor vortex, the turbulent kinetic energy is high, as it is demonstrated by experiment and simulation. But, the experiment shows only a small region of very high kinetic energy, whereas in the numerics nearly the complete floor vortex is of maximum intensity. This overestimation of kinetic energy must be a numerical artifact, which indicates that the resolution in this area should be further increased. Of course, the influence of the assumed turbulent intensity and length scale at the domain inlet can have an effect on this result as well. In the  $x_7$  plane, the accordance between PIV and LES-LBM is good, see Figs. 7.15(c) and 7.15(d). The shape of the kinetic distribution is well reproduced in the calculation, only the maximum values differ a bit to the experiment. A similar conclusion can be drawn for the  $y_3$  plane. Figures 7.15(e) and 7.15(f) agree excellently. The two remaining planes in Fig. 7.16 are for constant  $z$  coordinate. Both in the  $z_1$  and in the  $z_2$  plane, the turbulent kinetic energy is high in the region of the floor vortex. The simulation reproduces the energy distribution of the experiment very well.

Based on the velocity field, the vorticity can be calculated. Literature provides the experimental vorticity fields for the out-of-plane component in the  $z_1$  and  $z_2$  planes. The vorticity fields in these planes are compared to the numerical fields in Fig. 7.17, by means of the non-dimensional vorticity  $\omega^n$ . High vorticities only occur in the area of the floor vortex. In general, the conformity between experiment and simulation is good. In the  $z_1$  plane, the high vorticity region is more distinctive in the PIV results than in the calculation.

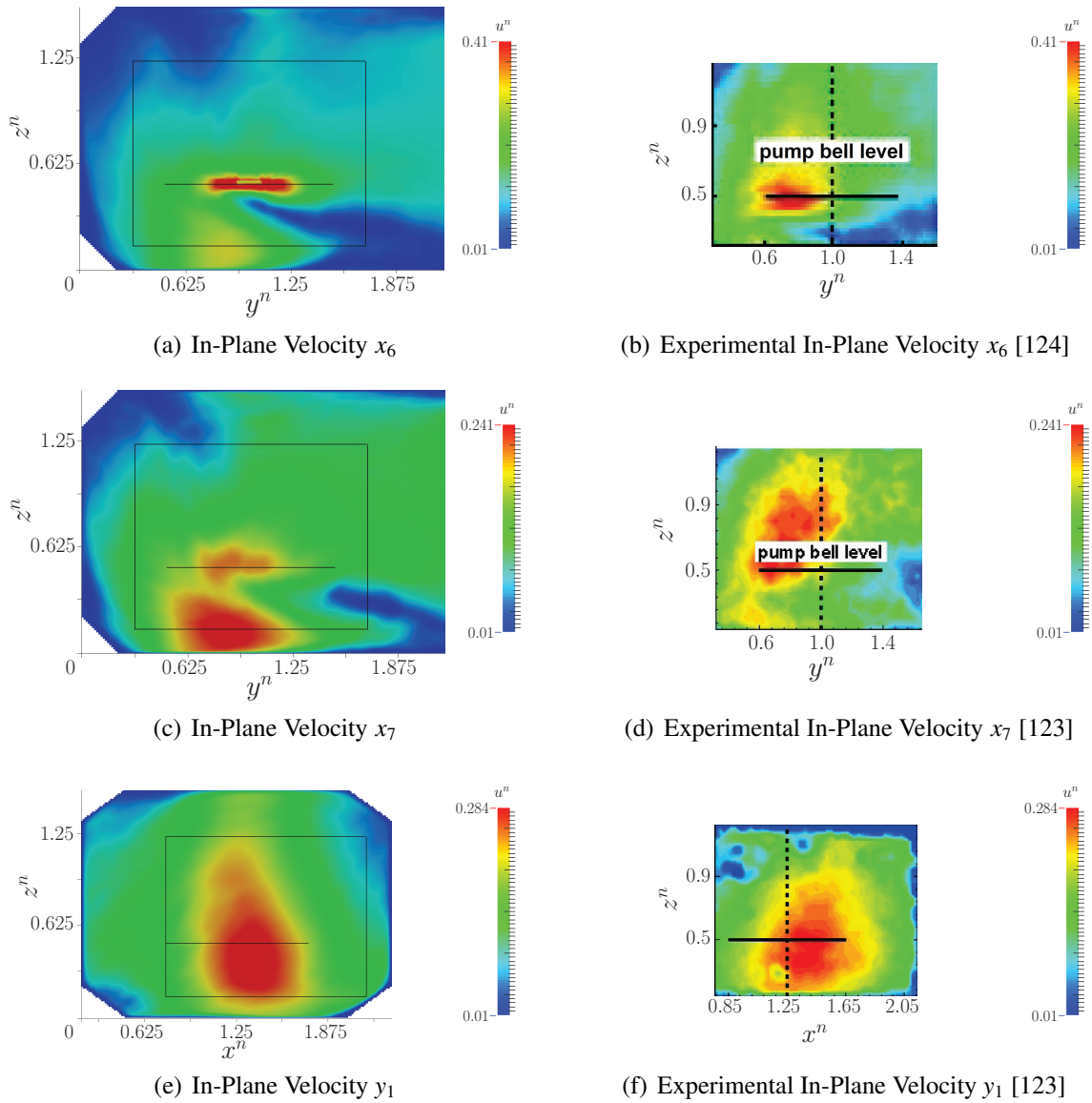


Figure 7.13: Time Averaged Absolute In-Plane Velocity [Reprint of figure (b) with permission from IIHR. Reprint of figures (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.]

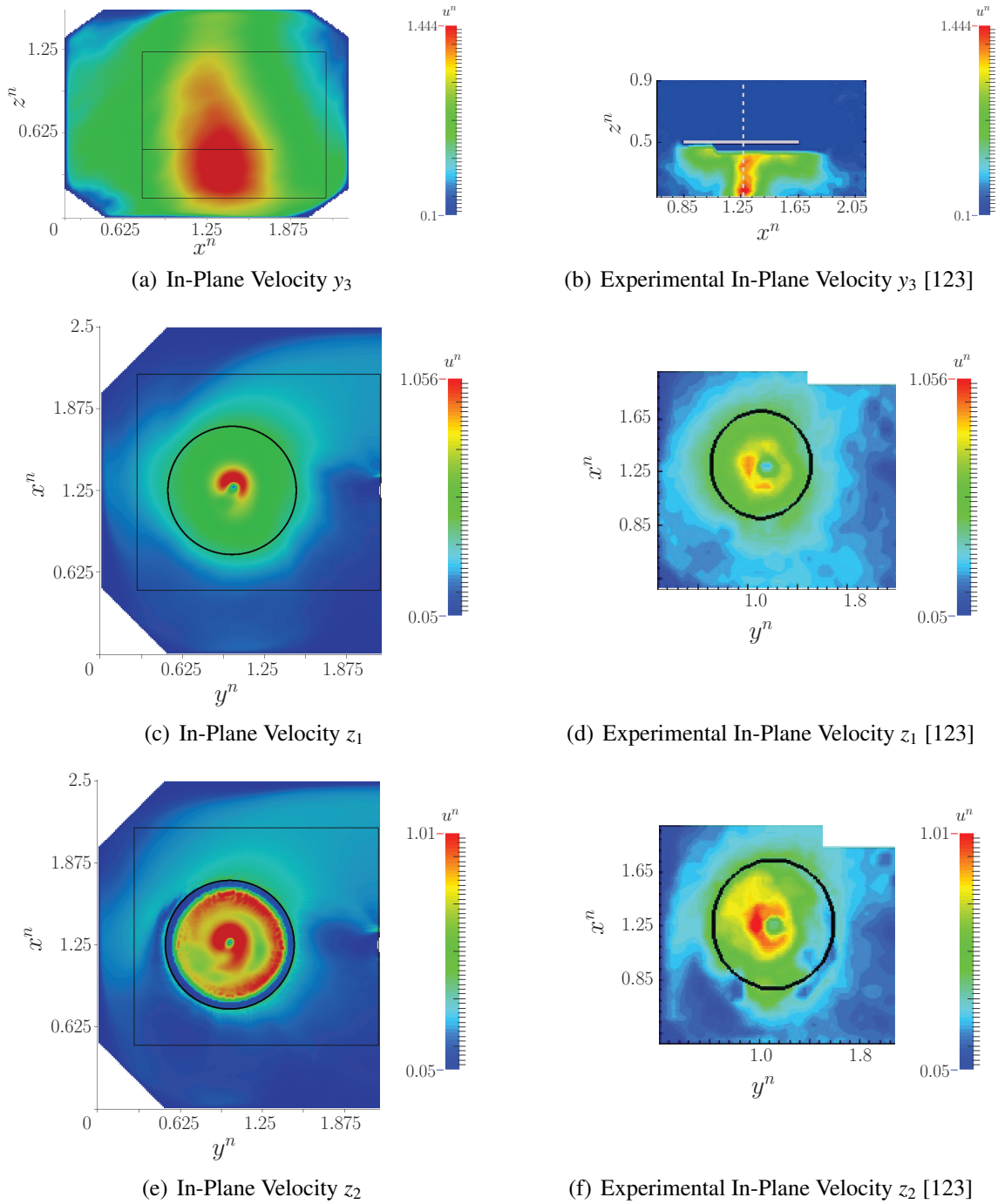
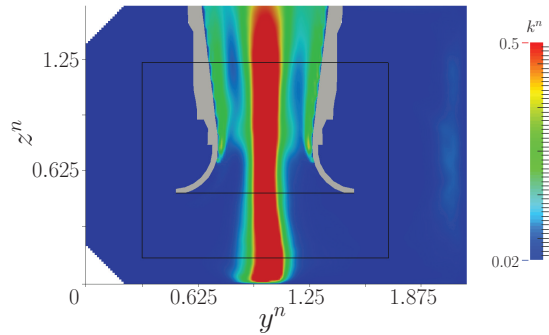
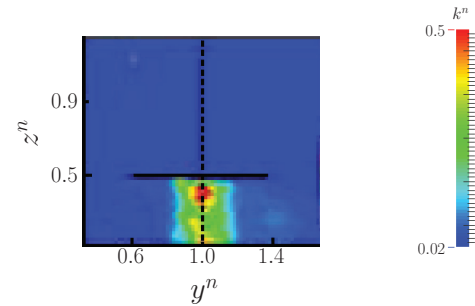


Figure 7.14: Time Averaged Absolute In-Plane Velocity (2) [Reprint of figures (b), (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.]

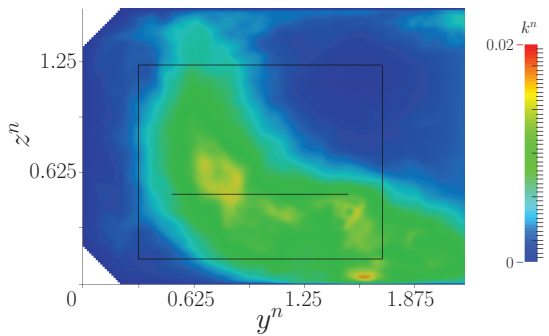




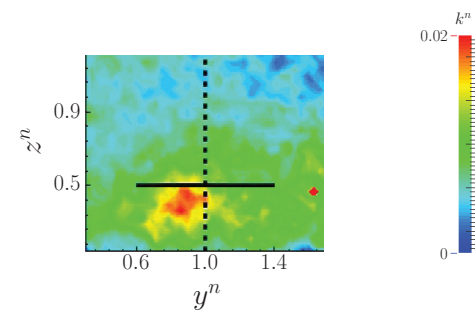
(a) In-Plane Turbulent Kinetic Energy  $x_4$



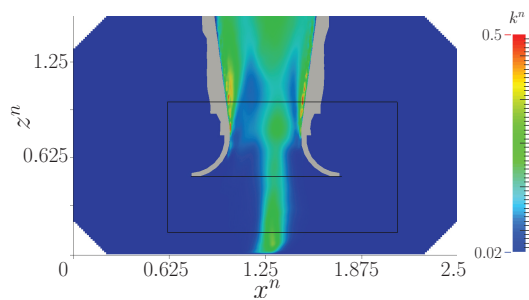
(b) Experimental In-Plane Turbulent Kinetic Energy  $x_4$  [124]



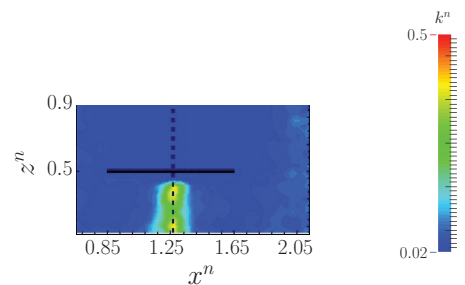
(c) In-Plane Turbulent Kinetic Energy  $x_7$



(d) Experimental In-Plane Turbulent Kinetic Energy  $x_7$  [123]



(e) In-Plane Turbulent Kinetic Energy  $y_3$



(f) Experimental In-Plane Turbulent Kinetic Energy  $y_3$  [123]

Figure 7.15: Time Averaged In-Plane Turbulent Kinetic Energy [Reprint of figure (b) with permission from IIHR. Reprint of figures (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.]

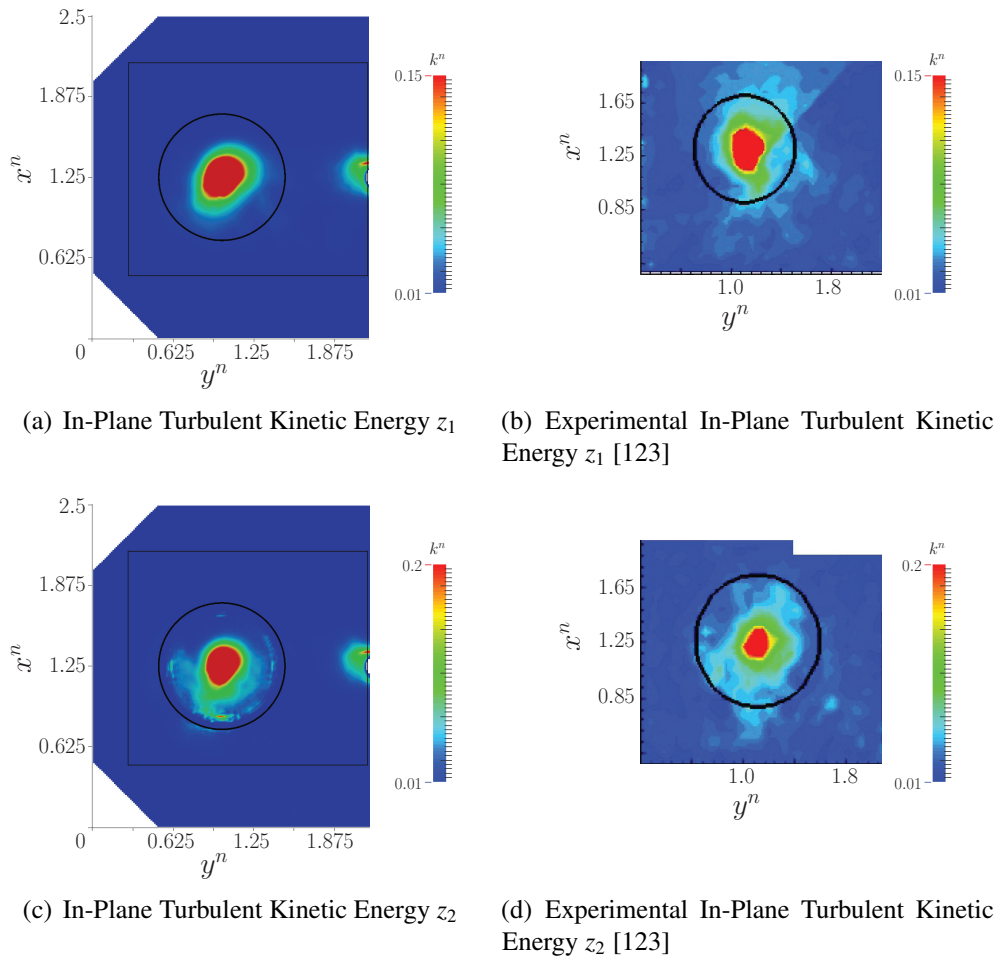


Figure 7.16: Time Averaged In-Plane Turbulent Kinetic Energy (2) [Reprint of figures (b) and (d) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.]

Finally, some pure quantitative comparisons are drawn on the basis of the velocity distribution and turbulent kinetic energy distribution along lines in some of the measurement planes. All experimental curves are taken from [122]. Figure 7.18(a) depicts the absolute in-plane velocity along a line in the  $y_3$  plane at position  $z^n = 0.284$ . The numerical solution is able to reproduce the shape of the velocity distribution of the experiment. Even more, the locations of the maximum velocities match very well. Although, there are obvious differences. The value of the maximum velocity in the simulation is remarkably smaller than in the experiment and also the regions around the velocity maximum are wider in the experiment. At a different line,  $z^n = 0.138$ , in the  $y_3$  plane the distribution of  $k$  is given by the experiment. A comparison with the simulation is shown in Fig. 7.18(b). Again, the characteristic curve shape is reproduced in



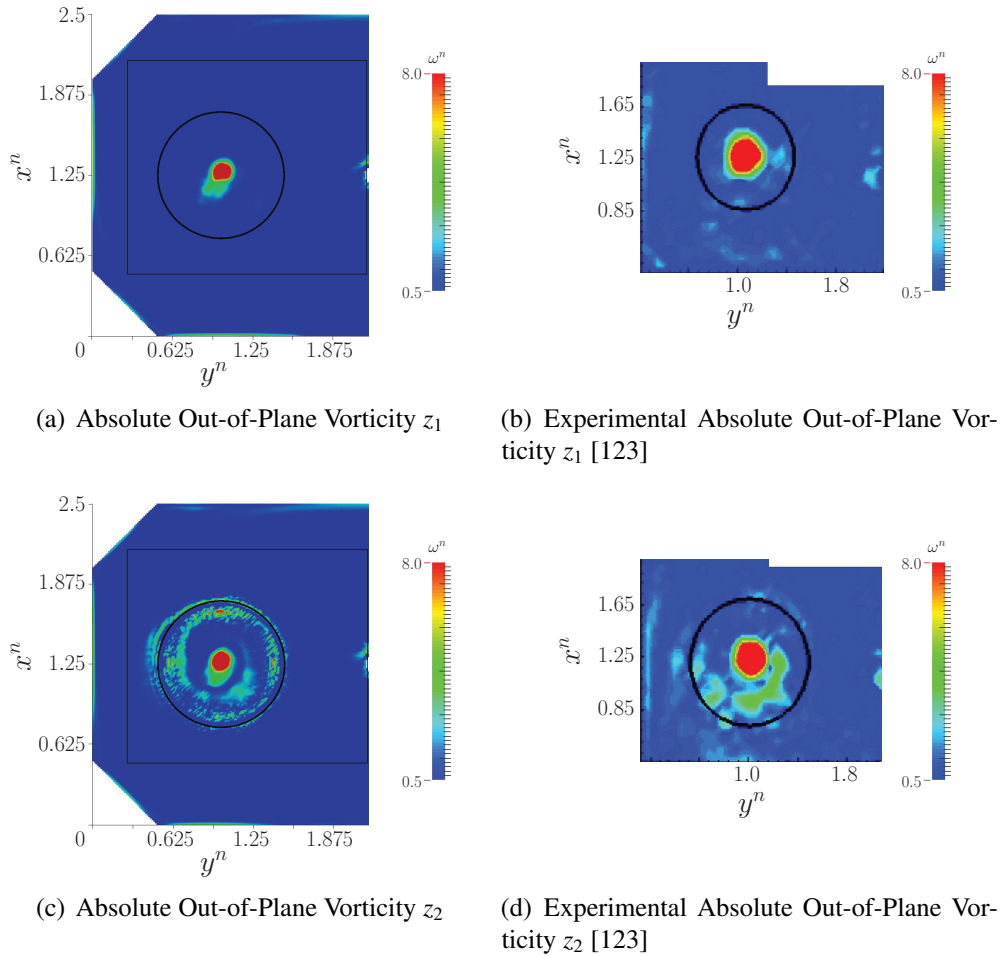


Figure 7.17: Time Averaged Out-of-Plane Vorticity [Reprint of figures (b) and (d) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.]

the calculation. The maximum locations differ slightly, even though the agreement is still good. As a comparison of the maximum values reveals, approximately 90% of the turbulent kinetic energy is resolved here. Thus, the LES is well resolved in this area.

Results are also available in the  $z_1$  plane for a line at  $x^n = 1.25$ , analogous to the distributions of  $u$  and  $k$  along the lines in the  $y_3$  plane. Figure 7.18(c) presents the numerical and PIV outcomes for the velocity along this line. The numerics capture the shape and the different velocity extrema of the experimental distribution, but the location of the extrema and their values differ. We see a similar behavior in the plot of the turbulent kinetic energy in Fig. 7.18(d). Qualitatively, the shape is reproduced by the simulation, but higher values for  $k$  are predicted. As said when analyzing the  $y_3$  plane, this is a numerical artifact.

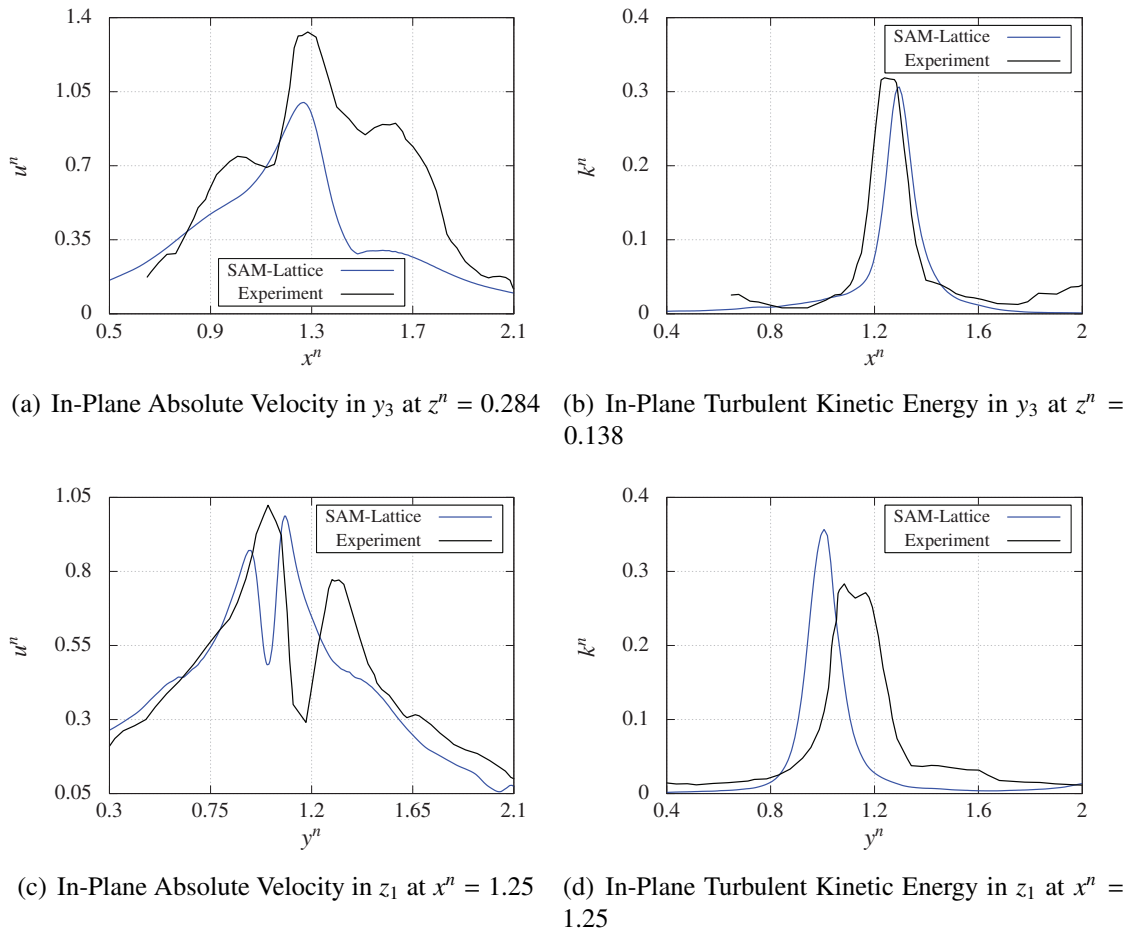


Figure 7.18: Quantitative Comparisons

A final assessment of the existing differences between the numerical results and the PIV measurements is hard to make, since the uncertainty of the measurements is not clearly specified in the according literature. Of course further calculations with different grids can be performed to investigate and eliminate the numerical artifact of overpredicting  $k$ .

The findings of this section clearly demonstrate the capabilities of the proposed LES-LBM, which is implemented in SAM-Lattice. The scheme enables a qualitative and quantitative representation of the very complex vortical flow in pump intakes. Besides the results in temporal average, even transient features of the flow field, like intermittency and meandering of vortices, are captured accurately and realistically.

---

### Conclusions and Future Work

---

In this thesis a consistent large eddy approach for Lattice Boltzmann Methods has been presented. The proposed approach includes, in addition to a subgrid scale model for consideration of the unresolved turbulent fluctuations, adequate boundary conditions. Wall treatment for turbulent flows is distinguished into wall resolved and wall modeled schemes. For wall resolved calculations, classical wall boundary conditions from the Lattice Boltzmann context are applied. In the case of wall modeled simulations, the effect of the unresolved part boundary layer must be considered appropriately. We have derived here a novel two layer wall model in the context of large eddy simulation for Lattice Boltzmann Methods. Besides wall boundaries, turbulent domain inlet boundary conditions are needed in most engineering flow problems. To create appropriate turbulent fluctuations at the inlet, a technique for synthetic turbulence, the synthetic eddy model, has been introduced into the Lattice Boltzmann Method.

For application to real world flow problems, the LES model has been implemented in the CFD package SAM-Lattice, which has been created in the course of this work. SAM-Lattice is capable of the calculation of incompressible or weakly compressible isothermal flows of engineering interest in arbitrary three dimensional domains. The modular design of the CFD package, the structure, and the principles of the different components were explained comprehensively. The basic principles of the Lattice Boltzmann Method, as it is used in SAM-Lattice, were documented in detail as well. This documentation has been performed in an educational sense, as it shall serve as a reference guide for students and researchers using SAM-Lattice. LBM was derived from kinetic theory, i.e., the Boltzmann equation, where the different assumptions and limitations made throughout this process were highlighted. In this way, possibilities to extent the method beyond its current limits are given.

Every development process of numerical methods requires verification and validation, which was conducted in this thesis. Verification of SAM-Lattice was performed for different analyt-

ical solutions of the Navier-Stokes equations in the laminar case. By means of Beltrami flow, Taylor-Couette flow, Couette flow, and Jeffery-Hamel flow, the different collision models and boundary conditions were proven. Results show the expected behavior: The program produces time accurate solutions for transient flows and reveals second order spatial accuracy, as far as the boundary conditions allow this. Also the applied grid refinement technique demonstrates its feasibility. The large eddy model and its components, respectively, were verified with help of turbulent channel flow, for which a large and confident DNS data basis is available. For the wall resolved cases, excellent agreement between the Lattice Boltzmann results and the reference solution was found. In the wall modeled example, also good accordance was detected. In combination with the successful prove of the synthetic eddy method, the accuracy and feasibility of the proposed LES model was shown.

To motivate new and broader applications of the Lattice Boltzmann Method in hydrodynamics, the suggested approach was validated for pump intake flows. Representative intake flows were considered for a free surface and a pressurized intake structure. In the case of the free surface intake, an additional modeling concept for the free surface was motivated and applied, since SAM-Lattice currently supports only single phase flows. Despite of this simplification, the proposed Lattice Boltzmann model is feasible to reproduce the multiple vortices in the free surface intake and the transient features of the vortices, like meandering and intermittency. Quantitative comparisons of the calculated and measured vortex positions showed excellent agreement. For the pressurized intake very detailed measurement results of velocity and turbulent kinetic energy fields are available. This allowed comprehensive qualitative and quantitative validations. As for the free surface intake, this second validation case revealed again that the large eddy Lattice Boltzmann approach is able to qualitatively and quantitatively represent the very complex vortical flow in the intake.

Some remarks for future work have already been given at the according positions in the text and will be summarized in the following. Considerable advantages are expected from a dynamic subgrid scale model within the LES. The dependency on the Smagorinsky constant can be removed in this way and also the capabilities of the scheme can be extended to transitional flows, by using an appropriate model. Instead of using a static wall function, the two layer approach could be improved by solving a simplified equation, e.g., the thin layer Navier-Stokes equations on a subgrid between the wall and the first fluid node. Another interesting and important topic is the further stabilization of the multi-level scheme to enable wall resolved calculations for higher Reynolds numbers, i.e., larger relaxation rates  $\Omega$ . A promising possibility may be the already mentioned filtering technique.

Additionally to these suggestions, which are direct deductions from the results and experiences of this thesis, the development and application of Lattice Boltzmann schemes for compressible and thermal flows are interesting, upcoming areas for LBM. As explained in this thesis, the physics for such flows can be treated in LBM by larger discrete velocity sets. One last field of application to mention here are rarefield gas flows, where conventional continuum approaches fail. Here, LBM can use its kinetic nature to go beyond the continuum limit.

## APPENDIX A

---

### D3Q19 MRT Model

---

Equilibrium Moments in Terms of Fluid Density and Velocity

$$\begin{aligned} m_0^{eq} &= \rho & m_1^{eq} &= \rho u^2 \\ m_3^{eq} &= \rho u_x & m_5^{eq} &= \rho u_y \\ m_7^{eq} &= \rho u_z & m_9^{eq} &= \rho (2u_x^2 - u_y^2 - u_z^2) \\ m_{11}^{eq} &= \rho (u_y^2 - u_z^2) & m_{13}^{eq} &= \rho u_x u_y \\ m_{14}^{eq} &= \rho u_y u_z & m_{15}^{eq} &= \rho u_x u_z \\ m_2^{eq} &= m_4^{eq} = m_6^{eq} = m_8^{eq} = m_{10}^{eq} = m_{12}^{eq} = m_{16}^{eq} = m_{17}^{eq} = m_{18}^{eq} = 0 \end{aligned} \quad (\text{A.1})$$



Relation between Macroscopic Stress Tensor  $\sigma$  and Moments  $m_i$

$$\begin{aligned}
 \sigma_{xx} &= \frac{\Omega\nu}{c_s^2\Delta t} \left( \frac{1}{3}m_1 + \frac{1}{3}m_9 - \frac{m_3^2}{m_0} \right) \\
 &= \frac{\Omega\nu}{c_s^2\Delta t} \left( \frac{1}{3}e + p_{xx} - \rho u_x^2 \right) \\
 \sigma_{xy} &= \frac{\Omega\nu}{c_s^2\Delta t} \left( m_{13} - \frac{m_3m_5}{m_0} \right) \\
 &= \frac{\Omega\nu}{c_s^2\Delta t} (p_{xy} - \rho u_x u_y) \\
 \sigma_{xz} &= \frac{\Omega\nu}{c_s^2\Delta t} \left( m_{15} - \frac{m_3m_7}{m_0} \right) \\
 &= \frac{\Omega\nu}{c_s^2\Delta t} (p_{xz} - \rho u_x u_z) \\
 \sigma_{yy} &= \frac{\Omega\nu}{c_s^2\Delta t} \left( \frac{1}{3}m_1 - \frac{1}{6}m_9 + \frac{1}{2}m_{11} - \frac{m_5^2}{m_0} \right) \\
 &= \frac{\Omega\nu}{c_s^2\Delta t} \left( \frac{1}{3}e - \frac{1}{2}p_{xx} + \frac{1}{2}p_{ww} - \rho u_y^2 \right) \\
 \sigma_{yz} &= \frac{\Omega\nu}{c_s^2\Delta t} \left( m_{14} - \frac{m_5m_7}{m_0} \right) \\
 &= \frac{\Omega\nu}{c_s^2\Delta t} (p_{yz} - \rho u_y u_z) \\
 \sigma_{zz} &= \frac{\Omega\nu}{c_s^2\Delta t} \left( \frac{1}{3}m_1 - \frac{1}{6}m_9 - \frac{1}{2}m_{11} - \frac{m_7^2}{m_0} \right) \\
 &= \frac{\Omega\nu}{c_s^2\Delta t} \left( \frac{1}{3}e - \frac{1}{2}p_{xx} - \frac{1}{2}p_{ww} - \rho u_z^2 \right)
 \end{aligned} \tag{A.3}$$

## Scaling Rules Coarse to Fine for MRT Scaling

$$\begin{aligned}
m_{15}^c &= \frac{\Omega^f \Delta t^c}{\Omega^c \Delta t^f} \left( m_{15}^f - \frac{m_3 m_7}{m_0} \right) + \frac{m_3 m_7}{m_0} = \frac{2\Omega^f}{\Omega^c} \left( m_{15}^f - \frac{m_3 m_7}{m_0} \right) + \frac{m_3 m_7}{m_0} \\
m_{14}^c &= \frac{\Omega^f \Delta t^c}{\Omega^c \Delta t^f} \left( m_{14}^f - \frac{m_5 m_7}{m_0} \right) + \frac{m_5 m_7}{m_0} = \frac{2\Omega^f}{\Omega^c} \left( m_{14}^f - \frac{m_5 m_7}{m_0} \right) + \frac{m_5 m_7}{m_0} \\
m_{13}^c &= \frac{\Omega^f \Delta t^c}{\Omega^c \Delta t^f} \left( m_{13}^f - \frac{m_3 m_5}{m_0} \right) + \frac{m_3 m_5}{m_0} = \frac{2\Omega^f}{\Omega^c} \left( m_{13}^f - \frac{m_3 m_5}{m_0} \right) + \frac{m_3 m_5}{m_0} \\
m_{11}^c &= \frac{\Omega^f \Delta t^c}{\Omega^c \Delta t^f} \left( m_{11}^f - \frac{m_5^2}{m_0} + \frac{m_7^2}{m_0} \right) + \frac{m_5^2}{m_0} - \frac{m_7^2}{m_0} = \frac{2\Omega^f}{\Omega^c} \left( m_{11}^f - \frac{m_5^2}{m_0} + \frac{m_7^2}{m_0} \right) + \frac{m_5^2}{m_0} - \frac{m_7^2}{m_0} \\
m_9^c &= 2 \frac{\Omega^f \Delta t^c}{\Omega^c \Delta t^f} \left( \frac{1}{2} m_9^f - \frac{1}{2} m_{11}^f - \frac{m_3^2}{m_0} + \frac{m_5^2}{m_0} \right) + m_{11}^c + \frac{2m_3^2}{m_0} - \frac{2m_5^2}{m_0} \\
&= \frac{4\Omega^f}{\Omega^c} \left( \frac{1}{2} m_9^f - \frac{1}{2} m_{11}^f - \frac{m_3^2}{m_0} + \frac{m_5^2}{m_0} \right) + m_{11}^c + \frac{2m_3^2}{m_0} - \frac{2m_5^2}{m_0} \\
m_1^c &= 3 \frac{\Omega^f \Delta t^c}{\Omega^c \Delta t^f} \left( \frac{1}{3} m_1^f + \frac{1}{3} m_9^f - \frac{m_3^2}{m_0} \right) - m_9^c + \frac{3m_3^2}{m_0} \\
&= \frac{6\Omega^f}{\Omega^c} \left( \frac{1}{3} m_1^f + \frac{1}{3} m_9^f - \frac{m_3^2}{m_0} \right) - m_9^c + \frac{3m_3^2}{m_0} \tag{A.4}
\end{aligned}$$

## Scaling Rules Fine to Coarse for MRT Scaling

$$\begin{aligned}
m_{15}^f &= \frac{\Omega^c \Delta t^f}{\Omega^f \Delta t^c} \left( m_{15}^c - \frac{m_3 m_7}{m_0} \right) + \frac{m_3 m_7}{m_0} = \frac{\Omega^c}{2\Omega^f} \left( m_{15}^c - \frac{m_3 m_7}{m_0} \right) + \frac{m_3 m_7}{m_0} \\
m_{14}^f &= \frac{\Omega^c \Delta t^f}{\Omega^f \Delta t^c} \left( m_{14}^c - \frac{m_5 m_7}{m_0} \right) + \frac{m_5 m_7}{m_0} = \frac{\Omega^c}{2\Omega^f} \left( m_{14}^c - \frac{m_5 m_7}{m_0} \right) + \frac{m_5 m_7}{m_0} \\
m_{13}^f &= \frac{\Omega^c \Delta t^f}{\Omega^f \Delta t^c} \left( m_{13}^c - \frac{m_3 m_5}{m_0} \right) + \frac{m_3 m_5}{m_0} = \frac{\Omega^c}{2\Omega^f} \left( m_{13}^c - \frac{m_3 m_5}{m_0} \right) + \frac{m_3 m_5}{m_0} \\
m_{11}^f &= \frac{\Omega^c \Delta t^f}{\Omega^f \Delta t^c} \left( m_{11}^c - \frac{m_5^2}{m_0} + \frac{m_7^2}{m_0} \right) + \frac{m_5^2}{m_0} - \frac{m_7^2}{m_0} = \frac{\Omega^c}{2\Omega^f} \left( m_{11}^c - \frac{m_5^2}{m_0} + \frac{m_7^2}{m_0} \right) + \frac{m_5^2}{m_0} - \frac{m_7^2}{m_0} \\
m_9^f &= 2 \frac{\Omega^c \Delta t^f}{\Omega^f \Delta t^c} \left( \frac{1}{2} m_9^c - \frac{1}{2} m_{11}^c - \frac{m_3^2}{m_0} + \frac{m_5^2}{m_0} \right) + m_{11}^f + \frac{2m_3^2}{m_0} - \frac{2m_5^2}{m_0} \\
&= \frac{\Omega^c}{\Omega^f} \left( \frac{1}{2} m_9^c - \frac{1}{2} m_{11}^c - \frac{m_3^2}{m_0} + \frac{m_5^2}{m_0} \right) + m_{11}^f + \frac{2m_3^2}{m_0} - \frac{2m_5^2}{m_0} \\
m_1^f &= 3 \frac{\Omega^c \Delta t^f}{\Omega^f \Delta t^c} \left( \frac{1}{3} m_1^c + \frac{1}{3} m_9^c - \frac{m_3^2}{m_0} \right) - m_9^f + \frac{3m_3^2}{m_0} \\
&= \frac{3\Omega^c}{2\Omega^f} \left( \frac{1}{3} m_1^c + \frac{1}{3} m_9^c - \frac{m_3^2}{m_0} \right) - m_9^f + \frac{3m_3^2}{m_0} \tag{A.5}
\end{aligned}$$



---

## Nomenclature

---

$\mathbf{\Pi}$	Momentum flux tensor
$\xi$	Velocity vector
$\delta$	Mean molecular spacing
$\Gamma$	Circulation
$\gamma$	Surface tension
$\kappa$	Wave number
$\lambda$	Free mean path
$\mu$	Dynamic viscosity
$\nu$	Kinematic viscosity
$\Omega$	Relaxation parameter
$\omega$	Collision frequency
$\overline{\Delta_c}$	Cutoff filter length
$\tau$	Collision time
$\tau_w$	Wall shear stress
$\mathbf{f}$	Force
$\mathbf{M}$	Transformation matrix
$\mathbf{m}$	Moment vector
$\mathbf{n}$	Normal vector
$\mathbf{q}$	Heat flux

---

<b>S</b>	Strain rate tensor
<b>u</b>	Fluid velocity
<b>x</b>	Position vector
$\zeta$	Bulk Viscosity
<i>a</i>	Acceleration
$C_s$	Smagorinsky constant
$c_s$	Speed of sound
$D, d$	Diameter
$d\xi$	Element of velocity space
$dN$	Number of molecules
$dV$	Volume element
<i>E</i>	Error
<i>e</i>	Internal energy
<i>f</i>	Probability distribution function
<i>Fr</i>	Froude number
<i>g</i>	Gravitational acceleration
<i>h</i>	Distance
<i>k</i>	Boltzmann constant
<i>Kn</i>	Knudsen number
<i>L</i>	Length
$l_m$	Mixing length
<i>M</i>	Moment of the distribution function
<i>m</i>	Molecular mass
<i>Ma</i>	Mach number
<i>n</i>	Particle density
<i>Nc</i>	Circulation number
<i>p</i>	Pressure
<i>Q</i>	Flow rate
$q_i$	Dimensionless wall distance

---

$R$	Gas constant
$Re$	Reynolds number
$S$	Entropy
$T$	Temperature
$t$	Physical time
$u_\tau$	Friction velocity
$w_i$	Weighting factor
$We$	Weber number
$y^+$	Dimensionless wall distance
API	Application programming interface
BREP	Boundary representation
CAX	Computer-aided technologies
CFD	Computational Fluid Dynamics
CGNS	CFD General Notation System
DNS	Direct numerical simulation
GUI	Graphical user interface
LBM	Lattice Boltzmann Method
LES	Large eddy simulation
MPI	Message passing interface
MRT	Multiple relaxation time model
ODE	Ordinary differential equation
OP	Operating point
RANS	Reynolds averaged Navier Stokes
SAT	Separating axis theorem
SEM	Synthetic eddy method
SMP	Shared memory parallelism
SRT	Single relaxation time model
V&V	Verification and validation
WM	Wall modeled
WR	Wall resolved



---

## List of Figures

---

2.1	Phase Space . . . . .	8
2.2	BBGKY Hierarchy . . . . .	14
3.1	Links to Boltzmann-BGK Equation . . . . .	21
3.2	D3Q19 Lattice . . . . .	23
3.3	Periodicity Condition . . . . .	30
3.4	Bounce Back Schemes . . . . .	31
3.5	Bouzidi Scheme . . . . .	33
3.6	Treatment for Curved Boundaries . . . . .	34
3.7	Interpolation for Shear Bounce Back Scheme . . . . .	37
3.8	Interpolation Scheme for Shear Anti Bounce Back Scheme . . . . .	38
3.9	Octree Refinement . . . . .	43
3.10	Simple Grid Layout (2D) . . . . .	43
3.11	Interface Structure (2D) . . . . .	44
3.12	Interface Structure (3D) . . . . .	45
3.13	Nested Time Stepping . . . . .	49
4.1	Scale Separation (based on [104]) . . . . .	54
4.2	Box Filter ( $\overline{\Delta_c} = 1$ ) . . . . .	56
4.3	Regions of a Turbulent Boundary Layer . . . . .	63
4.4	Stress Contributions in a Turbulent Boundary Layer . . . . .	63
4.5	Comparison of Log Law Approach . . . . .	66
4.6	Comparison of Power Law Approach . . . . .	67
4.7	Schematic Representation of a Precursor Simulation . . . . .	70
4.8	Box of Eddies . . . . .	72

5.1	Modules of SAM-Lattice . . . . .	78
5.2	Flowchart of SamGenerator . . . . .	79
5.3	Grid Refinement and Smoothing . . . . .	81
5.4	Flowchart of SamSolver . . . . .	85
6.1	Initialization Velocity Field ( $t = 0$ ) . . . . .	92
6.2	Comparison of Flow Quantities for Beltrami Flow . . . . .	93
6.3	Grid Convergence for Beltrami Flow . . . . .	94
6.4	Sketch of Taylor-Couette Flow . . . . .	95
6.5	Results for Taylor-Couette Flow . . . . .	97
6.6	Domain for Channel Flow . . . . .	99
6.7	Results for Plane Couette Flow . . . . .	101
6.8	Results for Annulus Couette Flow . . . . .	104
6.9	Domain for Jeffery-Hamel Flow . . . . .	105
6.10	Results for Jeffery-Hamel Flow . . . . .	107
6.11	Instantaneous Velocity Field . . . . .	110
6.12	Mean Velocity Profile . . . . .	111
6.13	Turbulent Velocity Fluctuations . . . . .	111
6.14	Turbulent Kinetic Energy Spectrum . . . . .	112
6.15	Instantaneous Velocity Field for Multi-Level Run . . . . .	113
6.16	Mean Velocity Profile for Multi-Level Run . . . . .	114
6.17	Turbulent Velocity Fluctuations for Multi-Level Run . . . . .	114
6.18	Turbulent Kinetic Energy Spectrum for Multi-Level Run . . . . .	115
6.19	Instantaneous Velocity Field for Wall Modeled Approach ( $C_s = 0.1$ ) . . . . .	116
6.20	Mean Velocity Profiles for Wall Modeled Approach . . . . .	117
6.21	Turbulent Velocity Fluctuations for Wall Modeled Approach ( $C_s = 0.1$ ) . . . . .	117
6.22	Mean Velocity Profiles for Different $y^+$ ( $C_s = 0.075$ ) . . . . .	118
6.23	Turbulent Kinetic Energy Spectrum for Wall Modeled Approach ( $C_s = 0.075$ ) . . . . .	119
6.24	Domain for SEM Test Case . . . . .	120
6.25	Results at Domain Inlet . . . . .	122
7.1	Exemplary Pump Intake and Geometrical Parameters . . . . .	124
7.2	Vortices in Pump Intakes . . . . .	126
7.3	3D Visualization of Vortex Structures at OP1 . . . . .	132
7.4	Time Averaged Surface Streamlines at OP1 . . . . .	133
7.5	3D Visualization of Vortex Structures at OP3 . . . . .	134
7.6	Time Averaged Surface Streamlines at OP3 . . . . .	135
7.7	Pressurized Pump Intake . . . . .	138
7.8	Turbulent Kinetic Energy Spectrum under Pump Center . . . . .	141
7.9	Time Averaged $Q$ -Criterion . . . . .	141

7.10	Time Averaged In-Plane Streamlines [Reprint of figures (b), (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.] . . . . .	143
7.11	Time Averaged In-Plane Streamlines (2) [Reprint of figures (b) and (d) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.] . . . . .	144
7.12	Time Averaged In-Plane Streamlines (3) [Reprint of figures (b), (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.] . . . . .	145
7.13	Time Averaged Absolute In-Plane Velocity [Reprint of figure (b) with permission from IIHR. Reprint of figures (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.] . . . . .	147
7.14	Time Averaged Absolute In-Plane Velocity (2) [Reprint of figures (b), (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.] . . . . .	148
7.15	Time Averaged In-Plane Turbulent Kinetic Energy [Reprint of figure (b) with permission from IIHR. Reprint of figures (d) and (f) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.] . . . . .	149
7.16	Time Averaged In-Plane Turbulent Kinetic Energy (2) [Reprint of figures (b) and (d) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.] . . . . .	150
7.17	Time Averaged Out-of-Plane Vorticity [Reprint of figures (b) and (d) with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.] . . . . .	151
7.18	Quantitative Comparisons . . . . .	152





---

## List of Tables

---

1.1	Mathematical Definitions . . . . .	5
6.1	Parameters for Beltrami Flow . . . . .	91
6.2	Mesh Dependent Parameters for Beltrami Flow . . . . .	94
6.3	Parameters for Taylor-Couette Flow . . . . .	96
6.4	Mesh Dependent Parameters for Taylor-Couette Flow . . . . .	98
6.5	Average Grid Convergence Rate for Taylor-Couette Flow . . . . .	98
6.6	Parameters for Plane Couette Flow . . . . .	100
6.7	Mesh Dependent Parameters for Plane Couette Flow . . . . .	102
6.8	Parameters for Annulus Couette Flow . . . . .	103
6.9	Mesh Dependent Parameters for Annulus Couette Flow . . . . .	103
6.10	Parameters for Jeffery Hamel Flow . . . . .	106
6.11	Parameters for Turbulent Channel Flow at $Re_\tau = 180$ . . . . .	109
6.12	Parameters for Turbulent Channel Flow at $Re_\tau = 180$ with Grid Refinement . . . . .	112
6.13	Parameters for Turbulent Channel Flow at $Re_\tau = 2000$ . . . . .	115
6.14	Mesh Dependent Parameters for Turbulent Channel Flow at $Re_\tau = 2000$ . . . . .	118
6.15	Parameters for SEM Test Case . . . . .	121
6.16	Turbulent Statics at Inlet Plane . . . . .	121
7.1	Types of Free Surface Vortices [66] . . . . .	125
7.2	Types of Subsurface Vortices [66] . . . . .	125
7.3	Dimensionless Parameters . . . . .	125
7.4	Geometrical Parameters of Free Surface Intake . . . . .	128
7.5	Free Surface Intake - Operating Points (OP) . . . . .	129
7.6	Simulation Parameters for Free Surface Intake . . . . .	130
7.7	Geometrical Parameters of Pressurized Intake . . . . .	137

7.8	Evaluation Planes . . . . .	139
7.9	Simulation Parameters for Pressurized Intake . . . . .	140

---

## Bibliography

---

- [1] T. Abe. Derivation of the Lattice Boltzmann Method by Means of the Discrete Ordinate Method for the Boltzmann Equation. *Journal of Computational Physics*, 131:241–246, 1997.
- [2] T. Akenine-Möller. Fast 3D triangle-box overlap testing. In *ACM SIGGRAPH 2005 Courses*, page 8. ACM, 2005.
- [3] American Society of Mechanical Engineers. Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer, 2009.
- [4] M. Ansar, T. Nakato, and G. Constantinescu. Numerical simulations of inviscid three-dimensional flows at single-and dual-pump intakes. *Journal of Hydraulic Research*, 40(4):461–470, 2002.
- [5] ANSI. American National Standard for Intake Design for Rotodynamic Pumps (ANSI/HI 9.8), 2012.
- [6] S. Ansumali and I. V. Karlin. Single relaxation time model for entropic lattice Boltzmann methods. *Phys. Rev. E*, 65:056312, May 2002.
- [7] *TGrid 5.0 User's Guide*. Ansys Inc., Centerra Resource Park, 10 Cavendish Court, Lebanon, NH 03766, USA, 2008.
- [8] E. Balaras, C. Benocci, and U. Piomelli. Two-layer approximate boundary conditions for large-eddy simulations. *AIAA journal*, 34(6):1111–1119, 1996.
- [9] P. L. Bhatnagar, E. P. Gross, and M. Krook. A Model For Collision Processes in Gases. *Phys. Rev.*, 94:511, 1954.
- [10] G. Bird. *Molecular Gas Dynamics*. Clarendon Press Oxford, 1976.
- [11] M. Bouzidi, M. Firdaouss, and P. Lallemand. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids (1994-present)*, 13(11):3452–3459, 2001.
- [12] R. Brownlee, A. N. Gorban, and J. Levesley. Stability and stabilization of the lattice Boltzmann method. *Physical Review E*, 75(3):036711, 2007.

- [13] R. A. Brownlee, J. Levesley, D. Packwood, and A. N. Gorban. Add-ons for Lattice Boltzmann Methods: Regularization, Filtering and Limiters. In M. Ehrhardt, editor, *Novel Trends in Lattice Boltzmann Methods: Reactive Flow, Physicochemical Transport and Fluid-Structure Interaction*. Bentham Science Publishers, 2013.
- [14] M. Burns. *Automated fabrication: improving productivity in manufacturing*. Prentice-Hall, Inc., 1993.
- [15] A. Caiazzo. *Asymptotic Analysis of lattice Boltzmann method for Fluid-Structure interaction problems*. PhD thesis, TU Kaiserslautern, 2007.
- [16] I. Celik, Z. Cehreli, and I. Yavuz. Index of resolution quality for large eddy simulations. *Journal of Fluids Engineering*, 127(5):949–958, 2005.
- [17] H. Chen and X. Shan. Fundamental conditions for N-th-order accurate lattice Boltzmann models. *Physica D: Nonlinear Phenomena*, 237:2003 – 2008, 2008.
- [18] H. Chen, C. Teixeira, and K. Molvig. Realization of Fluid Boundary Conditions via Discrete Boltzmann Dynamics. *International Journal of Modern Physics C*, 09(08):1281–1292, 1998.
- [19] H. Chen, O. Filippova, J. Hoch, K. Molvig, R. Shock, C. Teixeira, and R. Zhang. Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Physica A: Statistical Mechanics and its Applications*, 362(1):158–167, 2006.
- [20] H. Chen, I. Goldhirsch, and S. Orszag. Discrete Rotational Symmetry, Moment Isotropy, and Higher Order Lattice Boltzmann Models. *Journal of Scientific Computing*, 34(1): 87–112, 2008.
- [21] S. Chen and G. D. Doolen. Lattice Boltzmann Method For Fluid Flows. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1998.
- [22] D. Conrad. *A Viscosity Adaptive Lattice Boltzmann Method*. PhD thesis, TU Kaiserslautern, 2015.
- [23] G. Constantinescu and V. Patel. Numerical model for simulation of pump-intake flow and vortices. *Journal of Hydraulic Engineering*, 124(2):123–134, 1998.
- [24] G. Constantinescu and V. Patel. Role of turbulence model in prediction of pump-bay vortices. *Journal of Hydraulic Engineering*, 126(5):387–391, 2000.
- [25] B. Crouse. *Lattice-Boltzmann Strömungssimulationen auf Baumdatenstrukturen*. PhD thesis, TU München, 2003.
- [26] P. J. Dellar. Bulk and shear viscosities in lattice Boltzmann equations. *Phys. Rev. E*, 64: 031203, Aug 2001.
- [27] P. J. Dellar. Incompressible limits of lattice Boltzmann equations using multiple relaxation times. *Journal of Computational Physics*, 190(2):351 – 370, 2003.
- [28] P. J. Dellar. Lattice Boltzmann algorithms without cubic defects in Galilean invariance on standard lattices. *Journal of Computational Physics*, 259(0):270 – 283, 2014.
- [29] D. d’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transac-*

- tions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 360(1792):437–451, 2002.
- [30] Y.-H. Dong, P. Sagaut, and S. Marie. Inertial consistent subgrid model for large-eddy simulation based on the lattice Boltzmann method. *Physics of Fluids (1994-present)*, 20(3):035104, 2008.
- [31] A. Dupuis and B. Chopard. Theory and applications of an alternative lattice Boltzmann grid refinement algorithm. *Phys. Rev. E*, 67:066707, Jun 2003.
- [32] C. R. Ethier and D. Steinman. Exact fully 3D Navier–Stokes solutions for benchmarking. *International Journal for Numerical Methods in Fluids*, 19(5):369–375, 1994.
- [33] F. R. Feito and J. C. Torres. Inclusion test for general polyhedra. *Computers & Graphics*, 21(1):23–30, 1997.
- [34] J. H. Ferziger and M. Perić. *Computational methods for fluid dynamics*, volume 3. Springer Berlin, 1996.
- [35] O. Filippova and D. Hänel. Boundary-Fitting and Local Grid Refinement for Lattice-BGK Models. *International Journal of Modern Physics C*, 09(08):1271–1279, 1998.
- [36] O. Filippova, S. Succi, F. Mazzocco, C. Arrighetti, G. Bella, and D. Hänel. Multiscale lattice Boltzmann schemes with turbulence modeling. *Journal of Computational Physics*, 170(2):812–829, 2001.
- [37] S. Freudiger. *Entwicklung eines parallelen, adaptiven, komponentenbasierten Strömungskerns für hierarchische Gitter auf Basis des Lattice-Boltzmann-Verfahrens*. PhD thesis, TU Braunschweig, 2009.
- [38] J. Fröhlich and D. von Terzi. Hybrid LES/RANS methods for the simulation of turbulent flows. *Progress in Aerospace Sciences*, 44(5):349 – 377, 2008.
- [39] J. Fröhlich. *Large eddy simulation turbulenter Strömungen*. Springer, 2006.
- [40] S. E. Gant. Reliability issues of LES-related approaches in an industrial context. *Flow, turbulence and combustion*, 84(2):325–335, 2010.
- [41] E. Garnier, N. Adams, and P. Sagaut. *Large eddy simulation for compressible flows*. Springer, 2009.
- [42] M. Geier, A. Greiner, and J. G. Korvink. Bubble functions for the lattice Boltzmann method and their application to grid refinement. *The European Physical Journal Special Topics*, 171(1):173–179, 2009.
- [43] M. C. Geier. *Ab Initio Derivation of the Cascaded Lattice Boltzmann Automaton*. PhD thesis, University of Freiburg, 2006.
- [44] I. Ginzburg and D. d’Humières. Multireflection boundary conditions for lattice Boltzmann models. *Physical Review E*, 68(6):066614, 2003.
- [45] I. Ginzburg, F. Verhaeghe, and D. d’Humières. Two-relaxation-time lattice Boltzmann scheme: About parametrization, velocity, pressure and mixed boundary conditions. *Communications in computational physics*, 3(2):427–478, 2008.
- [46] Z. Guo and C. Shu. *Lattice Boltzmann Method and Its’ Applications in Engineering*.

- Advances in Computational Fluid Dynamics. World Scientific Publishing Company Incorporated, 2013.
- [47] Z. Guo, C. Zheng, and S. Baochang. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys Rev E*, 65:046308, 2002.
- [48] Z. Guo, C. Zheng, and B. Shi. Lattice Boltzmann equation with multiple effective relaxation times for gaseous microscale flow. *Phys. Rev. E*, 77:036707, Mar 2008.
- [49] S. M. Guzik, T. H. Weisgraber, P. Colella, and B. J. Alder. Interpolation methods and the accuracy of lattice-Boltzmann mesh refinement. *Journal of Computational Physics*, 259(0):461 – 487, 2014.
- [50] D. Hänel. *Molekulare Gasdynamik*. Springer, 2004.
- [51] T. Hübner. *A Monolithic, Off-Lattice Approach to the Discrete Boltzmann Equation with Fast and Accurate Numerical Methods*. PhD thesis, Technische Universität Dortmund, 2011.
- [52] X. He and L.-S. Luo. A priori derivation of the lattice Boltzmann equation. *Phys. Rev. E*, 55:R6333–R6336, Jun 1997.
- [53] X. He, Q. Zou, L.-S. Luo, and M. Dembo. Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model. *Journal of Statistical Physics*, 87(1-2):115–136, 1997.
- [54] X. He, S. Chen, and G. D. Doolen. A Novel Thermal Model for the Lattice Boltzmann Method in Incompressible Limit. *Journal of Computational Physics*, 146(1):282 – 300, 1998.
- [55] G. Hecker. *Swirling Flow Problems at Intakes*, chapter Fundamentals of vortex intake flow. IAHR Design Manual. CRC Press, 1987.
- [56] D. J. Holdych, D. R. Noble, J. G. Georgiadis, and R. O. Buckius. Truncation error analysis of lattice Boltzmann methods. *Journal of Computational Physics*, 193(2):595 – 619, 2004.
- [57] S. Hou, J. Sterling, S. Chen, and G. Doolen. A lattice Boltzmann subgrid model for high Reynolds number flows. *Pattern formation and lattice gas automata*, 6:151–166, 1996.
- [58] S. Hoyas and J. Jiménez. Scaling of the velocity fluctuations in turbulent channels up to  $Re_\tau = 2003$ . *Physics of Fluids (1994-present)*, 18(1):011702, 2006.
- [59] S. Izquierdo and N. Fueyo. Momentum transfer correction for macroscopic-gradient boundary conditions in lattice Boltzmann methods. *Journal of Computational Physics*, 229(7):2497 – 2506, 2010.
- [60] N. Jarrin. *Synthetic Inflow Boundary Conditions for the Numerical Simulation of Turbulence*. PhD thesis, School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, 2008.
- [61] N. Jarrin, S. Benhamadouche, D. Laurence, and R. Prosser. A synthetic-eddy-method for generating inflow conditions for large-eddy simulations. *International Journal of Heat and Fluid Flow*, 27(4):585–593, 2006.

- [62] J. Jiménez and P. Moin. The minimal flow unit in near-wall turbulence. *Journal of Fluid Mechanics*, 225:213–240, 1991.
- [63] M. Junk, A. Klar, and L.-S. Luo. Asymptotic analysis of the lattice Boltzmann equation. *Journal of Computational Physics*, 210(2):676–704, 2005.
- [64] H. S. Kang, S. Chester, and C. Meneveau. Decaying turbulence in an active-grid-generated flow and comparisons with large-eddy simulation. *Journal of Fluid Mechanics*, 480:129–160, 2003.
- [65] U. Khan, N. Ahmed, Z. Zaidi, S. Jan, and S. T. Mohyud-Din. On Jeffery-Hamel Flows. *Int. J. Modern Math. Sci*, 7(3):236–247, 2013.
- [66] K. Kirst. *Experimentelle und numerische Untersuchungen von Zulaufbedingungen vertikaler Pumpsysteme*. PhD thesis, TU Kaiserslautern, 2012.
- [67] M. Krafczyk, J. Tölke, and L.-S. Luo. Large-eddy simulations with a multiple-relaxation-time LBE model. *International Journal of Modern Physics B*, 17(01n02):33–39, 2003.
- [68] T. Krüger, F. Varnik, and D. Raabe. Second-order convergence of the deviatoric stress tensor in the standard Bhatnagar-Gross-Krook lattice Boltzmann method. *Phys. Rev. E*, 82:025701, Aug 2010.
- [69] A. Ladd and R. Verberg. Lattice-Boltzmann Simulations of Particle-Fluid Suspensions. *Journal of Statistical Physics*, 104(5-6):1191–1251, 2001.
- [70] A. J. C. Ladd. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *Journal of Fluid Mechanics*, 271:285–309, 7 1994.
- [71] D. Lagrava, O. Malaspinas, J. Latt, and B. Chopard. Advances in multi-domain lattice Boltzmann grid refinement. *Journal of Computational Physics*, 231(14):4808 – 4822, 2012.
- [72] P. Lallemand and L.-S. Luo. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E*, 61:6546–6562, Jun 2000.
- [73] P. Lallemand and L.-S. Luo. Theory of the lattice Boltzmann method: Acoustic and thermal properties in two and three dimensions. *Phys. Rev. E*, 68:036706, Sep 2003.
- [74] J. Latt. *Hydrodynamic limit of lattice Boltzmann equations*. PhD thesis, Université de Genève, 03/09 2007.
- [75] M. Lesieur. *Turbulence in fluids*. Springer, 2008.
- [76] E. Lévêque, F. Toschi, L. Shao, and J.-P. Bertoglio. Shear-improved Smagorinsky model for large-eddy simulation of wall-bounded turbulent flows. *Journal of Fluid Mechanics*, 570:491–502, 2007.
- [77] Y. Li, R. Shock, R. Zhang, and H. Chen. Numerical study of flow past an impulsively started cylinder by the lattice-Boltzmann method. *Journal of Fluid Mechanics*, 519:273–300, 11 2004.
- [78] C. Lucino and S. Gonzalo Dur. Vortex detection in pump sumps by means of CFD. In

- XXIV latin American congress on hydraulics*, Punta del Este, Uruguay, 2010.
- [79] L.-S. Luo, W. Liao, X. Chen, Y. Peng, and W. Zhang. Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations. *Physical Review E*, 83(5):056710, 2011.
- [80] O. Malaspinas and P. Sagaut. Advanced large-eddy simulation for lattice Boltzmann methods: The approximate deconvolution model. *Physics of Fluids (1994-present)*, 23(10):105103, 2011.
- [81] O. Malaspinas and P. Sagaut. Consistent subgrid scale modelling for lattice Boltzmann methods. *Journal of Fluid Mechanics*, 700:514–542, 2012.
- [82] O. P. Malaspinas. *Lattice Boltzmann method for the simulation of viscoelastic fluid flows*. PhD thesis, STI, Lausanne, 2009.
- [83] S. Marié, D. Ricot, and P. Sagaut. Comparison between lattice Boltzmann method and Navier-Stokes high order schemes for computational aeroacoustics. *Journal of Computational Physics*, 228(4):1056 – 1070, 2009.
- [84] R. Mei, L.-S. Luo, and W. Shyy. An Accurate Curved Boundary Treatment in the Lattice Boltzmann Method. *Journal of Computational Physics*, 155(2):307 – 330, 1999.
- [85] R. Mei, W. Shyy, D. Yu, and L.-S. Luo. Lattice Boltzmann method for 3-D flows with curved boundary. *Journal of Computational Physics*, 161(2):680–699, 2000.
- [86] R. Mei, D. Yu, W. Shyy, and L.-S. Luo. Force evaluation in the lattice Boltzmann method involving curved geometry. *Physical Review E*, 65(4):041203, 2002.
- [87] R. Mei, L.-S. Luo, P. Lallemand, and D. d’Humières. Consistent initial conditions for lattice Boltzmann simulations. *Computers & Fluids*, 35(8-9):855 – 862, 2006.
- [88] R. D. Moser, J. Kim, and N. N. Mansour. Direct numerical simulation of turbulent channel flow up to  $Re_\tau = 590$ . *Phys. Fluids*, 11(4):943–945, 1999.
- [89] X. Nie, X. Shan, and H. Chen. Galilean invariance of lattice Boltzmann models. *EPL (Europhysics Letters)*, 81(3):34005, 2008.
- [90] C. J. Ogayar, R. J. Segura, and F. R. Feito. Point in solid strategies. *Computers & Graphics*, 29(4):616–624, 2005.
- [91] T. Okamura, K. Kamemoto, J. Matsui, et al. CFD prediction and model experiment on suction vortices in pump sump. In *Proceedings of the 9th Asian International Conference on Fluid Machinery*, Jeju, Korea, 2007.
- [92] C. Pantano, R. Deiterding, D. Hill, and D. Pullin. A low numerical dissipation patch-based adaptive mesh refinement method for large-eddy simulation of compressible flows. *Journal of Computational Physics*, 221(1):63 – 87, 2007.
- [93] U. Piomelli. Wall-layer models for large-eddy simulations. *Progress in aerospace sciences*, 44(6):437–446, 2008.
- [94] S. B. Pope. *Turbulent flows*. Cambridge university press, 2001.
- [95] K. N. Premnath and J. Abraham. Three-dimensional multi-relaxation time (MRT) lattice-Boltzmann models for multiphase flow. *Journal of Computational Physics*, 224(2):539 –



- 559, 2007.
- [96] K. N. Premnath, M. J. Pattison, and S. Banerjee. Dynamic subgrid scale modeling of turbulent flows using lattice-Boltzmann method. *Physica A: Statistical Mechanics and its Applications*, 388(13):2640–2658, 2009.
- [97] K. N. Premnath, M. J. Pattison, and S. Banerjee. An investigation of the lattice Boltzmann method for large eddy simulation of complex turbulent separated flow. *Journal of Fluids Engineering*, 135(5):051401, 2013.
- [98] Y. Qian, D. D’Humières, and P. Lallemand. Lattice BGK Models for Navier-Stokes Equation. *Europhysics Letters*, 17:479–484, 1992.
- [99] P. Quéméré, P. Sagaut, and V. Couailler. A new multi-domain/multi-resolution method for large-eddy simulation. *International journal for numerical methods in fluids*, 36(4):391–416, 2001.
- [100] M. Rheinländer. A consistent grid coupling method for lattice-Boltzmann schemes. *Journal of Statistical Physics*, 121(1-2):49–74, 2005.
- [101] D. Ricot, S. Marié, P. Sagaut, and C. Bailly. Lattice Boltzmann method with selective viscosity filter. *Journal of Computational Physics*, 228(12):4478 – 4490, 2009.
- [102] M. Rohde, D. Kandhai, J. Derksen, and H. Van den Akker. A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes. *International journal for numerical methods in fluids*, 51(4):439–468, 2006.
- [103] F. Roman, V. Armenio, and J. Fröhlich. A simple wall-layer model for large eddy simulation with immersed boundary method. *Physics of Fluids (1994-present)*, 21(10):101701, 2009.
- [104] P. Sagaut. *Large eddy simulation for incompressible flows*. Springer, 2006.
- [105] P. Sagaut. Toward advanced subgrid models for Lattice-Boltzmann-based Large-eddy simulation: theoretical formulations. *Computers & Mathematics with Applications*, 59(7):2194–2199, 2010.
- [106] H. Schlichting and K. Gersten. *Boundary-layer theory*. Springer, 2000.
- [107] A. Schneider, D. Conrad, and M. Böhle. Numerical simulation of the flow field in pump intakes by means of Lattice Boltzmann methods. *IOP Conference Series: Materials Science and Engineering*, 52(2):022028, 2013.
- [108] A. Schneider, D. Conrad, and M. Böhle. Lattice Boltzmann Simulation of the Flow Field in Pump Intakes - A New Approach. *Journal of Fluids Engineering*, 137(3):031105, 2015.
- [109] X. Shan and H. Chen. A general multiple-relaxation-time Boltzmann collision model. *International Journal of Modern Physics C*, 18(04):635–643, 2007.
- [110] X. Shan and X. He. Discretization of the Velocity Space in the Solution of the Boltzmann Equation. *Phys. Rev. Lett.*, 80:65–68, Jan 1998.
- [111] X. Shan, X.-F. Yuan, and H. Chen. Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation. *Journal of Fluid Mechanics*, 550:413 – 441,

- 2006.
- [112] A. Shapiro. The use of an exact solution of the Navier-Stokes equations in a validation test of a three-dimensional nonhydrostatic numerical model. *Monthly weather review*, 121(8):2420–2425, 1993.
- [113] J. Smagorinsky. General Circulation Experiments with the Primitive Equations. *Monthly Weather Review*, 91 (3):99–164, 1963.
- [114] J. D. Sterling and S. Chen. Stability Analysis of Lattice Boltzmann Methods. *Journal of Computational Physics*, 123(1):196 – 206, 1996.
- [115] S. Stolz and N. A. Adams. An approximate deconvolution procedure for large-eddy simulation. *Physics of Fluids (1994-present)*, 11(7):1699–1701, 1999.
- [116] S. Succi. *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford university press, 2001.
- [117] M. Tagomori and M. Gotoch. Flow patterns and vortices in pump-sumps. In *Proc, Int Symp on Large Hydr Machinery, China press, Beijing, China*, pages 13–22, 1989.
- [118] X. L. Tang, F. J. Wang, Y. J. Li, G. H. Cong, X. Y. Shi, Y. L. Wu, and L. Y. Qi. Numerical investigations of vortex flows and vortex suppression schemes in a large pumping-station sump. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 225(6):1459–1480, 2011.
- [119] C. M. Teixeira. Incorporating turbulence models into the lattice-Boltzmann method. *International Journal of Modern Physics C*, 9(08):1159–1175, 1998.
- [120] L. Temmerman, M. A. Leschziner, C. P. Mellen, and J. Fröhlich. Investigation of wall-function approximations and subgrid-scale models in large eddy simulation of separated flow in a channel with streamwise periodic constrictions. *International Journal of Heat and Fluid Flow*, 24(2):157–180, 2003.
- [121] M. Terracol. A Zonal RANS/LES Approach for Noise Sources Prediction. *Flow, Turbulence and Combustion*, 77(1-4):161–184, 2006.
- [122] T. Tokyay and S. Constantinescu. Validation of a large-eddy simulation model to simulate flow in pump intakes of realistic geometry. *Journal of Hydraulic Engineering*, 132 (12):1303–1315, 2006.
- [123] T. E. Tokyay and S. G. Constantinescu. *Large Eddy Simulation and Reynolds Averaged Navier-Stokes Simulations of Flow in a Realistic Pump Intake: A Validation Study*, chapter 423, pages 1–12.
- [124] T. E. Tokyay and S. G. Constantinescu. Investigation of Flow Physics of Pump Intake Flows using Large Eddy Simulation. IIHR Technical Report No. 445, IIHR - Hydroscience & Engineering, The University of Iowa, 2005.
- [125] J. Tölke and M. Krafczyk. Second order interpolation of the flow field in the lattice Boltzmann method. *Computers & Mathematics with Applications*, 58(5):898–902, 2009.
- [126] J. Tölke, S. Freudiger, and M. Krafczyk. An adaptive scheme using hierarchical grids for lattice Boltzmann multi-phase flow simulations. *Computers & fluids*, 35(8):820–830,

- 2006.
- [127] F. Tosi, S. Ubertini, S. Succi, H. Chen, and I. V. Karlin. A Comparison of Single-Time Relaxation Lattice Boltzmann Schemes with Enhanced Stability. *International Journal of Modern Physics C*, 17(10):1375–1390, 2006.
  - [128] H. Touil, D. Ricot, and E. L ev eque. Direct and large-eddy simulation of turbulent flows on composite multi-resolution grids by the lattice Boltzmann method. *Journal of Computational Physics*, 256(0):220 – 233, 2014.
  - [129] H. Werner and H. Wengle. Large-eddy simulation of turbulent flow over and around a cube in a plate channel. In *Turbulent Shear Flows 8*, pages 155–168. Springer, 1993.
  - [130] F. White. *Viscous Fluid Flow*. McGraw-Hill Education, 1991.
  - [131] D. C. Wilcox. *Turbulence modeling for CFD*. DCW industries La Canada, CA, 1994.
  - [132] D. A. Wolf-Gladrow. *Lattice-gas cellular automata and lattice Boltzmann models: an introduction*. Number 1725. Springer, 2000.
  - [133] Y. Wu, Y. Li, and X. Li. PIV experiments on flow in model suction sump. Technical report, Thermal Engineering Department, Tsinghua University, 2000.
  - [134] Y. Wu, Y. Li, and X. Li. Measurements of Flow and Submergend Vortices in a Model Pump Suction Sump. Technical report, Thermal Engineering Department, Tsinghua University, 2001.
  - [135] D. Yu and S. S. Girimaji. Multi-block Lattice Boltzmann method: Extension to 3D and validation in turbulence. *Physica A: Statistical Mechanics and its Applications*, 362(1): 118–124, 2006.
  - [136] D. Yu, R. Mei, and W. Shyy. A multi-block lattice Boltzmann method for viscous fluid flows. *International Journal for Numerical Methods in Fluids*, 39(2):99–120, 2002.
  - [137] D. Yu, R. Mei, and W. Shyy. A Unified Boundary Treatment in Lattice Boltzmann Method. *New York: AIAA*, 953, 2003.



# Andreas Schneider

## Curriculum Vitae

### Work Experience

- 06/2009–Present    Research Assistant  
Institute of Fluid Mechanics and Fluid Machinery (SAM)  
University of Kaiserslautern, Germany
- 05/2008–08/2008    Internship in Compressor Aerodynamics Department  
MTU Aero Engines GmbH, Munich, Germany
- 11/2006–06/2009    Student Assistant  
Institute of Fluid Mechanics and Fluid Machinery (SAM)  
University of Kaiserslautern, Germany

### Education

- 10/2004–06/2009    Diploma (Dipl.-Ing.) in Mechanical Engineering  
University of Kaiserslautern, Germany  
Majors: Fluid Mechanics, Turbo Machinery, Thermodynamics  
Thesis: Erstellung eines Lattice-Boltzmann Algorithmus zur Simulation  
von Gitterströmungen
- 08/1995–03/2004    Allgemeine Hochschulreife (High School Diploma)  
Sickingen Gymnasium Landstuhl, Germany  
Majors: Mathematics, Physics, Politics