Simulation of Complex Flows and Multi-Physics with the Lattice-Boltzmann Method

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Universiteit van Amsterdam op gezag van de Rector Magnificus prof. dr. D. C. van den Boom ten overstaan van een door het college voor promoties ingestelde commissie, in het openbaar te verdedigen in de Agnietenkapel op donderdag 17 januari 2008, te 14:00 uur

door

Jörg Matthias Bernsdorf

geboren te Erlangen, Duitsland

Promotiecommissie:

Promotor:	prof. dr. P. M. A. Sloot
Co-promotor:	dr. ir. A. G. Hoekstra
O verige leden:	prof. dr. E. T. van Bavel
	prof. dr. D. Frenkel
	prof. dr. M. Krafczyk
	dr. G. Lonsdale
	dr. D. R. Hose
Faculteit:	Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The work described in this thesis has been carried out at the Section Computational Science of the University of Amsterdam, the Institute of Fluid Mechanics of the University of Erlangen-Nuremberg and the C&C Research Laboratories (now: IT Research Division, NEC Laboratories Europe), NEC Europe Ltd., St.Augustin.

The financial support by the following institutions is gratefully acknowledged:

Bayerische Forschungsstiftung in the Bavarian Consortium of High–Performance Scientific Computing (FORTWIHR II)

Deutsche Forschungsgemeinschaft (Proj.Nr.Br 1864/1)

@neurIST Integrated Project, which is co-financed by the European Commission through the contract no. IST-027703

IST Project COAST, which is co-financed by the European Commission through the contract no. 033664

Typeset with $\bot T_E X 2_{\mathcal{E}}$

Author contact: joergb@science.uva.nl j.bernsdorf@it.neclab.eu joerg@bernsdorf.org For my daughter Julia, who kindly offered her help when she found me working late at night. Boltzmann's work met with mixed reactions during his lifetime, and continues to do so even today.

Stanford Encyclopedia of Philosophy

As such models become more sophisticated and realistic, more traditional numerical methods of simulating fluid flow in porous media and fractured rock (...) lose their competitive edge, and will be phased out in the future.

Muhammad Sahimi

Reaching full maturity is just a matter of time and labour, no conceptual hurdles in sight.

Sauro Succi

Contents

1	Intr	oduction	1
	1.1	Complex Flows and Multi-Physics	1
	1.2	Computational Fluid Dynamics	3
		1.2.1 CFD-I: Navier-Stokes	3
		1.2.2 CFD-II: Lattice-Boltzmann	3
	1.3	Research Focus	4
		1.3.1 Structure of the Thesis	4
2	The	Method	7
	2.1	Cellular Automata: A Short History	7
	2.2	From Particles to Fluids	9
		2.2.1 Lattice-Gas (FHP-I)	9
		2.2.2 Lattice-Boltzmann	12
		2.2.3 Fluid-Solid Boundary Conditions	13
		2.2.4 Geometry Discretisation	14
		2.2.5 From the Boltzmann Equation to Lattice-Boltzmann	19
3	Imp	lementation Challenges	25
	3.1	Performance Optimised Implementation	26
		3.1.1 Full Matrix versus Sparse Implementation	26
		3.1.2 Performance Measurement on a Vector Computer	27
		3.1.3 Detailed Analysis: The Medical Case	31
	3.2	MPI-Parallelisation	33
		3.2.1 Partitioning	34
		3.2.2 MPI Performance Results	38
	3.3	Effort-Efficiency Considerations	40
		3.3.1 Choosing the Hardware and Programming Language	41
		3.3.2 Latest Trends	42
	3.4	Visualisation	43
		3.4.1 Concurrent Visualisation System RVSLIB	44
4	Con	nplex-Flow Studies	47
	4.1	Channel Flow in Increasingly Complex Geometries	47
		4.1.1 Conclusion	51
	4.2	Decay of Turbulence	51
		4.2.1 Time Evolution of a Vortex	51

		4.2.2	Decaying Turbulence Field	. 53
		4.2.3	Conclusion	. 55
	4.3	Plain C	Channel Flow around a Square Cylinder	. 56
		4.3.1	The Flow Problem	. 56
		4.3.2	Finite-Volume Method	. 58
		4.3.3	Details of the Test Case	. 59
		4.3.4	Results and Discussion	. 62
		4.3.5	Conclusion	. 71
	4.4	Numer	rical Analysis of the Pressure Drop in Porous Media Flow	. 72
		4.4.1	Analytical Models for the Pressure Drop	. 73
		4.4.2	Experimental Results	. 75
		4.4.3	Numerical Simulation of Porous Media Flow	. 76
		4.4.4	Conclusion	. 83
5	Mul	ti-Physi	ics Extensions	85
	5.1	Chemi	cal Reaction Modelling	. 85
	5.2	Nonlin	ear Adsorption / Desorption	. 87
		5.2.1	Lattice-Boltzmann Extension: Adsorption Model	. 87
		5.2.2	Simulation Results	. 87
		5.2.3	Visualisation	. 89
		5.2.4	Conclusion	. 90
	5.3	Concu	rrent Numerical Simulation of Flow and Clotting	. 90
		5.3.1	Medical Background	. 91
		5.3.2	Lattice-Boltzmann Extension: Aging Model	. 92
		5.3.3	Lattice-Boltzmann Extension: Clotting Model	. 93
		5.3.4	Simulation Results	. 93
		5.3.5	Conclusion	. 99
6	Disc	ussion,	Vision and Outlook	101
	6.1	Softwa	are Development	. 103
	6.2	Perspe	ctives of the Lattice Boltzmann Method for Industrial Applications	. 105
		6.2.1	Application Strategies	. 105
		6.2.2	Benefits of Industrial Simulations for the LB Community	. 105
	6.0	6.2.3	and for the Industry	. 106
	6.3	Blood	Flow Simulation in Cerebral Aneurysms: A Lattice-Boltzmann	100
		Medic	al Physics Application within the @neurIST Project	. 106
		6.3.1	Medical Background	. 10/
		6.3.2	Image Segmentation	. 108
		0.3.3		. 108
		6.3.4	Flow Simulation	. 109
	61	0.3.3 M:-14	Simulation Kesuits	. 109
	0.4	IVIUIti-	Scale Multi-Science Simulation: A Lattice-Boltzmann Medical Physic	CS
	65	Applic	vation within the COAST Project	. 112
	0.3			. 113
		0.3.1	Things I did not do	. 113
		0.3.2		. 110

Bibliography	117
Samenvatting	135
Acknowledgments	137
Publications	139

List of Figures

2.1	Hexagonal Bravais lattice (one lattice node is highlighted).	10
2.2	Lattice node with particles at cells 1, 2 and 5	10
2.3	Flow diagram of the cellular automata algorithm.	10
2.4	Propagation: particles moving to new positions.	11
2.5	Head-on and three particle collisions.	11
2.6	Acceleration: randomly chosen particles change their velocity vector	11
2.7	Mean flow rate of plane Poiseuille flow vs. iteration number for FHP and	
	LBA	12
2.8	Flow diagram illustrating the voxel-mesh generation from real objects	15
2.9	Digitised electron microscope picture of a sedimentary layer	16
2.10	2D Lattice-Boltzmann simulation of flow in a sedimentary layer	17
2.11	Flow diagram illustrating the voxel-mesh generation from CAD data	17
2.12	Streamlines around an IC engine and turbulent flow field around an ASMO	
	shape	18
2.13	Flow diagram illustrating the voxel-mesh generation from synthetic ge-	
	ometries	18
2.14	Flow through the solid phase created by spinodal decomposition	19
2.15	Stencil of a D3Q19 lattice.	21
31	Mapping of fluid nodes onto a graph and construction of the adjacency list	27
3.2	Benchmark cases 2-12	29
3.3	Comparison of full matrix and sparse Lattice-Boltzmann performance	_/
0.0	(MFLOP/s).	30
3.4	Comparison of full matrix and sparse Lattice-Boltzmann speed (MLUP/s).	31
3.5	Comparison of full matrix and sparse Lattice-Boltzmann memory con-	-
	sumption (MBYTE).	32
3.6	Geometry case 11: medical image showing an abdominal aorta with an	
	aneurysm in the iliac region.	32
3.7	Partitioning of two complex geometries for a sparse Lattice-Boltzmann	
	code by METIS: aorta and porous media.	36
3.8	Three partitions of a computational domain mapped on a graph. The out-	
	going and incoming links from neighbouring partitions are identified	36
3.9	Three partitions of a computational domain mapped on a graph. The out-	
	going links are copied into a ghost-layer during the propagation step	36
3.10	Three partitions of a computational domain mapped on a graph. The in-	
	coming links are copied from the receive buffer	37

Three partitions of a computational domain mapped on a graph. Inner nodes, exposed nodes and the receive-buffer are identified	37
Flow-chart of an MPI-parallelised Lattice-Boltzmann algorithm with over- lapping computation at inner nodes and communication at the boundaries.	38
Performance of an MPI-parallelised Lattice-Boltzmann algorithm for three different geometries on the NEC SX-8	39
Performance of an MPI-parallelised Lattice-Boltzmann algorithm for three different geometries on the INTEL Xeon cluster.	39
RVSLIB GUI and a snapshot from a time dependent visualisation of the chemical reaction $A + B \rightleftharpoons C$	45
Channel dimensions and locations for evaluating the pressure difference	48
Obstacle structure with increasing complexity	48
Profile of velocity magnitude in a cross section closely behind the obstacle area for different obstacle numbers.	49
Pressure drop and maximum velocity versus grid size using Lattice-Boltzmann	
and Finite-Volume methods.	50
Error in pressure drop and maximum velocity versus number of obstacles	
using Lattice-Boltzmann and Finite-Volume	50
Square velocity and velocity vectors of a vortex	53
Time evolution of vorticity and square velocity of a vortex	53
Time evolution of the vorticity of a homogeneous isotropic turbulent field	
for 0, 0.5, 2 and 4 eddy turn-over times. \ldots \ldots \ldots \ldots \ldots	54
Time evolution of dissipation and turbulent kinetic energy for a homoge- neous isotropic turbulent flow field	55
Definition of the geometry and integration domain.	60
Streamlines around the square cylinder for different Reynolds numbers.	63
Streak-lines around the square cylinder for different Reynolds numbers	64
Computed recirculation length vs. Reynolds number.	65
Computed drag coefficient vs. Reynolds number for steady flow	66
Comparison of instantaneous Lattice-Boltzmann and Finite-Volume results at a certain moment: stream-wise and cross-stream velocities along the	
centreline.	66
Comparison of instantaneous Lattice-Boltzmann and Finite-Volume results	
at a certain moment: stream-wise and cross-stream velocities at three dif-	
ferent positions in the flow field, centre of cylinder, near-wake and far-wake.	67
Computed Strouhal numbers vs. Reynolds number for Lattice-Boltzmann and Finite-Volume on different grids	68
Comparison of computed Stroubal numbers (Lattice-Boltzmann and Finite-	00
Volume) on finest grids with data from the literature	69
Computed time-averaged drag coefficient vs. Reynolds number for un-	07
steady flow.	70
Variation of force coefficient vs. Reynolds number for Finite-Volume results.	71
Friction coefficient versus Reynolds number	75
Packed bed of spheres.	77
Friction coefficient versus particle diameter.	77
	Three partitions of a computational domain mapped on a graph. Inner nodes, exposed nodes and the receive-buffer are identified Flow-chart of an MPI-parallelised Lattice-Boltzmann algorithm with overlapping computation at inner nodes and communication at the boundaries Performance of an MPI-parallelised Lattice-Boltzmann algorithm for three different geometries on the NEC SX-8

4.24 Computer tomography data as input for the Lattice-Boltzmann simulation	: 70
4.25 Decouver field for the flow through a catalytic converter	
4.25 Pressure field for the flow through a catalytic converter	. /8
4.26 Computer tomography data as input for the Lattice-Boltzmann simulation	: 70
	79
4.27 Pressure field for the flow through a porous SiC matrix.	. 80
4.28 Friction factor as a function of Reynolds number and relation of elongation	1
and shear, both for five different geometries and six different Reynolds	8
numbers	82
4.29 Dissipation by shear, by elongation; sum and total dissipation.	82
4.30 Correlation between tortuosity and relationship of elongation and shear.	83
5.1 Snapshot from a time dependent visualisation of the chemical reaction	1
$A + B \rightleftharpoons C$. 86
5.2 Link-wise concentration of the adsorbed species s_n at the solid surface, and	1
solvent concentration c in the fluid phase.	. 88
5.3 Linear plot of the time evolution (first 45,000 iterations) of the concentra-	-
tion of the adsorbed and dissolved fractions for an exponent $n = p = 0.8$.	. 89
5.4 Logarithmic plot of the time evolution of the concentration of the adsorbed	1
and dissolved fractions for the exponents $n = p = 0.5$ and $n = p = 0.8$.	. 89
5.5 Screen snapshot of the visualisation. Displayed is the concentration of the	e
solvent (x-y plane) and the adsorbed species (on the surface of the porous	5
media) at two different time steps.	. 90
5.6 Angiographic image of a stenosed coronary artery.	92
5.7 Clotting is modelled by increasing the solid fraction when the concentra-	-
tion of the tracer species reaches a threshold.	. 93
5.8 Idealised 75% stenosis geometry.	. 94
5.9 Comparison of experimental and numerical results for flow through a 75%)
stenosis at $Re = 100$.	. 95
5.10 Snapshot of a velocity iso surface of fully developed turbulent flow through	ı
a 75% stenosis at $Re = 550$. 95
5.11 Comparison of simulated and experimental flow through a 75% stenosis a	t
Re = 550 at four different times.	. 96
5.12 Tracer concentration ('age') for parabolic channel flow.	. 96
5.13 Growing clot downstream of a 2D stenosis at different time steps	. 97
5.14 Secondary milk clot in the experiment and secondary peak in the traces	r
concentration of the numerical simulation.	. 98
5.15 Comparison of an experimentally produced milk clot with the Lattice	-
5.15 Comparison of an experimentally produced milk clot with the Lattice Boltzmann simulation.	-
 5.15 Comparison of an experimentally produced milk clot with the Lattice Boltzmann simulation. 5.16 3D milk clot downstream of a stenosis at <i>Re</i> = 100. 	-
 5.15 Comparison of an experimentally produced milk clot with the Lattice Boltzmann simulation. 5.16 3D milk clot downstream of a stenosis at <i>Re</i> = 100. 6.1 Tetrahedral mesh of a cerebral Aneurysm and voxel mesh of the Lattice. 	-
 5.15 Comparison of an experimentally produced milk clot with the Lattice Boltzmann simulation. 5.16 3D milk clot downstream of a stenosis at <i>Re</i> = 100. 6.1 Tetrahedral mesh of a cerebral Aneurysm and voxel mesh of the Lattice Boltzmann simulation. 	- 98 99 - 108
 5.15 Comparison of an experimentally produced milk clot with the Lattice Boltzmann simulation. 5.16 3D milk clot downstream of a stenosis at <i>Re</i> = 100. 6.1 Tetrahedral mesh of a cerebral Aneurysm and voxel mesh of the Lattice Boltzmann simulation. 6.2 Streamlines indicating the complex flow inside the cerebral aneurysm. 	- 98 99 - 108 108
 5.15 Comparison of an experimentally produced milk clot with the Lattice Boltzmann simulation. 5.16 3D milk clot downstream of a stenosis at <i>Re</i> = 100. 6.1 Tetrahedral mesh of a cerebral Aneurysm and voxel mesh of the Lattice Boltzmann simulation. 6.2 Streamlines indicating the complex flow inside the cerebral aneurysm. 6.3 Shear stress distribution at an x-z cutting plane inside the cerebral aneurysm. 	- 98 99 - 108 108 110 110

6.5	Wall shear stress distribution at $Re = 1$	111
6.6	Wall shear stress distribution at $Re = 120$	112
6.7	Scale-map for in-stent re-stenosis.	113

List of Tables

3.1	Dimensions of the 12 benchmark cases.	28
3.2	Comparison of performance, speed and memory-consumption between full	
	matrix and sparse implementation.	30
3.3	Comparison of sparse and full matrix performance for a medical geometry	
	(case 11)	33
4.1	Overview of all grids used.	60

Chapter 1

Introduction

1.1 Complex Flows and Multi-Physics

The various phenomena of fluid flow have such presence in our daily life, that most human beings are fascinated by their complexity from early childhood. 'Experiments' on surface wave propagation (age 1), water channeling (age 5) or wind experiments with any kind of flying objects (age 7) help us to develop a natural feeling and understanding of complex flow phenomena before any scientific analysis.

Conversely, the impressive complexity and often seemingly unpredictable behaviour of fluids leads to many kinds of surprises in daily life, which are hard to explain without deeper scientific understanding: the suddenly increasing amount of ketchup coming out of the bottle (shear thinning fluids); the unexpected terrible noise when opening the window of a badly designed car (Helmholtz cavitation); the various sounds air-flow can cause when interacting with solid objects (vortex shedding and fluid-structure interaction at the onset of transient or turbulent flow), finding its most artistic expression in a rich variety of musical wind-instruments all over the world.

Nowadays, a deep insight into the dynamics of fluid flow is required for the efficient design of many kinds of technical devices, from micrometer scale (reading heads of the harddisc) to hundreds of meters size (large ships or buildings). Beyond the design of technical devices, research related to geological (e.g. ground water flow) and biological (e.g. blood flow) phenomena requires an understanding of the underlying fluid behaviour.

Often, a solution of the flow problem alone is insufficient to explain certain complex phenomena. It is the interaction of the flow with other physical, chemical or biological processes which has to be considered within a coupled scheme, to achieve the understanding required for modelling and simulation. If this is the case, we speak of 'multi-physics'.

The aforementioned fluid-structure interaction within musical instruments is an example of multi-physics flow, since coupling of the flow field to structure mechanics must be considered.¹

¹One might argue that for wind instruments (reed instruments excluded, where the fluid-structure interaction is obvious) like flutes or trumpets only the geometry is relevant for a certain steady sound-wave to establish in the corpus of the instrument. Alas, as the author can confirm from his own experience studying

More complex multi-physics flow problems in daily life are the 'aging' of a catalytic converter in a car,² or the complex biophysical processes in the human body which prevent blood clotting inside the arteries and veins, but ensure rapid healing of open wounds.³

The prediction of flow phenomena, aside from the intuitive (and often misleading) understanding as described above, is obtained from the triangle of experiments, analytical solutions of the underlying equations and numerical simulations.

Since analytical solutions of the governing equations are only possible for relatively simple boundary conditions, with a few degrees of freedom in a framework of simplified, reduced expressions, wind, water or oil-tunnel experiments have accompanied previous major developments. These experiments are usually extremely time-consuming and require large-scale highly specific equipment and elaborated methods to acquire precise data. For this reason, experimental fluid dynamics is very expensive and the production of results requires significant experience, time and forward planning.⁴ Nevertheless, and against the explicit announcements of some prestigious automotive companies, experimental investigations are still the backbone of most developments in this field.

The increasing availability of relatively cheap computer power, together with an equivalent development of efficient numerical methods during the past fifty or so years, allows for the simulation of complex fluid flow systems with many million degrees of freedom. The advantages of computational fluid dynamics (CFD) compared to experiments are obvious: general-purpose hardware can be used for simulations with standard commercial CFD-software and results can be obtained in a relatively short time allowing detailed insight into flow phenomena, sometimes not accessible by experiments.

The difficulty of setting up and running large expensive experimental equipment is shifted to the challenge of efficiently implementing numerical methods to solve the governing equations, and in the case of complex multi-physics phenomena, to perform the appropriate modelling and mathematical description of the phenomena that are the subject of a simulation.

the Shakuhachi [1], this is not the case. The interaction of the air pressure oscillations inside the instrument with the surrounding material seem to play an important role for the formation of the sound, which the musicians sometimes describe as 'vibration' of their instrument (some interesting remarks on this and other aspects of Shakuhachi physics can be found in [2]). A Lattice-Boltzmann simulation of a flute (neglecting the fluid-structure interaction) has been carried out by Kühnelt at the Institut für Wiener Klangstil [3, 4].

 $^{^{2}}$ This is related to catalytic chemical reactions in complex geometries (Chap. 5.1) and adsorption/resorption processes (Chap. 5.2).

 $^{^{3}}$ A first study on clotting processes will be presented in Chap. 5.3.

⁴The author had the chance to make his own experiences with the complicated process of acquiring experimental wind-tunnel data with the help of LDA [5] and heat-wire devices [6], during his time as a research scientist at LSTM, Erlangen.

1.2 Computational Fluid Dynamics

1.2.1 CFD-I: Navier-Stokes

The most common approach in CFD is to discretise the Navier-Stokes equation, which governs the flow properties of a 'simple fluid':⁵

$$\vec{\nabla}(\rho\vec{v}) = 0, \tag{1.1}$$

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v}\vec{\nabla})\vec{v} = -\frac{1}{\rho}\vec{\nabla}p + \frac{\mu}{\rho}\Delta\vec{v}$$
(1.2)

where \vec{v} is the velocity vector, ρ is the fluid density, μ is the dynamic viscosity and p is the pressure.

Almost 50 years of experience with hundreds of thousands of person years of development resulted in a rich diversity of highly sophisticated CFD codes, including many commercially available packages. Nevertheless, carrying out a CFD simulation has remained an expert's task, often with unknown predictability of success or estimation of the quality of the data.⁶

The major paradigm in the development of this 'classical' CFD approach was to invent more and more sophisticated methods of implementing the same set of equations. Alas, often in science progress is not made from repeatedly improving the same idea, but from stepping back and solving the problem from a completely different point of view.

So, it was a lucky coincidence, that theoretical physicists, who were developing 'toymodels' for a better understanding of non-equilibrium statistical mechanics, realised that their approach could actually be used to solve real-world flow problems. That is, how the Lattice-Gas and later Lattice-Boltzmann method emerged.⁷

1.2.2 CFD-II: Lattice-Boltzmann

Instead of solving the homogeneous (Navier-Stokes) equation on the macroscopic level, in the microscopic (Lattice-Gas) or mesoscopic (Lattice-Boltzmann) approach a set of equations derived from statistical physics is considered. Simply, the idea behind Lattice-Boltzmann is not to look at the time and space-development of the fluid, but at the (average) momentum and interaction of its particles. In contrast to the numerical simulation of the Boltzmann equation itself, this is done for a simplified (time and space-discrete) scheme.

Within the framework of a simple Lattice-Boltzmann simulation, an equidistant orthogonal lattice is chosen.⁸ On every lattice node, a set of real numbers, the particle density distributions, is stored. Updating of the lattice consists of two steps: a streaming process, where the

⁵For a 'simple fluid' we consider transport coefficients to be independent of the flow properties (Newtonian fluid) with no phase transition occurring.

⁶In not just a few cases the 'C' in 'CFD' is more obviously the abbreviation for 'Coloured' than anything related to engineering or science.

⁷For a short history of these so called 'cellular automata' see Chap. 2.1.

⁸More advanced approaches allow local mesh refinement, first suggested by Filippova and Hänel [7] or

particle densities are shifted in discrete time steps through the lattice along the connection lines to their next neighbouring nodes, and a relaxation step, where locally, a new particle distribution is computed by evaluating an equivalent to the Boltzmann collision integrals.

Every time step, the flow variables present in the Navier-Stokes equation (velocity, density) can be locally computed in terms of moments of this density distribution, while the viscosity is a function of the relaxation constant (equivalent to the collision frequency of the particles) and the pressure is derived via an equation of state from the density.⁹

Since the numerical solution of these equations simulates a fluid governed by the Navier-Stokes equation, it is sometimes argued that this is not a particularly exciting idea. The true excitement the method caused came from providing a simpler set of equations, which can generally be implemented in a more efficient way than the Navier-Stokes equation, resulting in higher performance.¹⁰

1.3 Research Focus

By reviewing a variety of examples from the past ten years of the author's research, undertaken within various co-operations, the focus behind this thesis is two-fold: first, giving a contribution to fluid mechanics and multi-physics research, for which Lattice-Boltzmann was used as a powerful tool to perform the numerical simulations; and second, to demonstrate the applicability of the Lattice-Boltzmann method as a competitive flow solver, in terms of performance and applicability to real-world flow problems.

For problems where traditional CFD (Navier-Stokes) approaches have difficulties due to the handling of complex or varying boundaries, or adding extension for the simulation of complex multi-physics procedures, the advantages of the Lattice-Boltzmann method are demonstrated.

The choice of the method and its particular implementation within this research was driven by the application. Certain features and hardware-specific optimisations were only implemented if required to solve the problem, and not simply because they were possible.

1.3.1 Structure of the Thesis

In Chapter 2 the method is introduced together with a short historical review of the development and, for didactic reason, a basic explanation of the Lattice-Gas approach. The step to Lattice-Boltzmann and a derivation of the Lattice-Boltzmann equation from the Boltzmann equation conclude this chapter. It is the intention of the second chapter to give the reader from *outside* the CFD/Lattice-Boltzmann community an understanding of the method.

octree-based data structures as suggested by Krafczyk [8]. Local mesh refinement is not considered in this thesis.

⁹Details of the method will be explained in Chap. 2.2.2.

¹⁰By 'performance' we speak of the relation between theoretical peak performance of the hardware in question and the performance a code achieves on this hardware. This is not always coming along with a shorter turn-around time to solve the flow problem itself. Performance aspects are discussed in Chap. 3.1.

A complete introduction and derivation of the governing equations can be found in a variety of text-books [9, 10, 11, 12, 13].

Chapter 3 reviews basic concepts required for the efficient implementation of a Lattice-Boltzmann solver. A detailed performance analysis, comparing a full-matrix and a sparse Lattice-Boltzmann code, illustrate the necessity to apply advanced algorithms and data structures for achieving the required performance to address 'real world' applications. The aspect of MPI-parallelisation is briefly discussed, and an introduction to a specific approach for the visualisation of HPC simulations concludes this chapter.

Chapter 4 presents several detailed quantitative flow studies, investigating steady- and transient complex flows in a variety of geometries. After initial validations of the method in comparison with analytical and other numerical results, the problem of pressure loss in complex geometries is analysed. The detailed simulations of porous media flow, made possible with the Lattice-Boltzmann method, demonstrated the 'tortuosity' as being overestimated within the common 'capillary theories' and the deformation of fluid elements is identified as a second dissipative source, which is usually neglected within these theories. These numerical simulations confirmed experimental simulations carried out by Durst et al. [14] in the mid eighties, which were at the time heavily criticised.

Aside from a contribution to fluid mechanics research, the intention of this chapter is to highlight the Lattice-Boltzmann method as an efficient and competitive tool, when applied to the simulation of transient or complex flows within complex geometries.

In Chapter 5, various extensions to the flow solver for modelling multi-physics phenomena are presented. It is demonstrated how from relatively simple local rules (making use of the cellular-automata-like structure behind Lattice-Boltzmann) a surprisingly complex behaviour emerges on the macro-scale. By presenting qualitative and quantitative simulations from various research fields, such as heterogeneous catalytic reaction, adsorption and milk/blood clotting, this chapter is dedicated to illustrating the strength of the Lattice-Boltzmann method as a complex-flow multi-physics simulation tool.

The last Chapter 6 briefly reviews Lattice-Boltzmann software development and provides some arguments for the industry and researchers to engage in addressing 'real world' problems with Lattice-Boltzmann. A brief outline of the author's current research within two European projects in the area of medical physics concludes this thesis.

Due to the large variety of studies presented within this thesis, not everything could be explained in full detail. The reader is referred to the author's publications on which the various chapters are based. Since the work of almost ten years is accumulated – ten years, in which the Lattice-Boltzmann method rapidly developed – not all results presented in the following have been carried out using today's state-of-the-art implementations.¹¹

Some of the older studies were, at the time of publication, the first of their kind within the Lattice-Boltzmann community, and although achieved with what are now slightly out of date implementations, nevertheless present valid results.

¹¹See also Chap. 6.5.1 'If I did it today ...'.

As explained in Chapter 3.1, more elaborate methods than a full matrix implementation are available today, which have been applied to the most recent simulations in the area of medical physics, presented in the Chapters 5.3, 6.3 and 6.4. These studies are ongoing research and partially leading in their field even outside the small scope of Lattice-Boltzmann.¹²

The author strongly believes in the future of the Lattice-Boltzmann method, with which he was concerned for the past ten years of academic and industrial research. An additional target of this thesis is therefore to promote the Lattice-Boltzmann method (with various extensions) as a very useful and highly efficient simulation tool for the academic and industrial study of complex-flow multi-physics problems.

¹²A few recently submitted collaborative papers on milk and blood clotting have not been included in this thesis, since they are subject of the PhD thesis of S.E.Harrison at the University of Sheffield, in the supervision of which the author was involved. For details, the reader is referred to Sarah's very comprehensive thesis [15].

Chapter 2

The Method

This chapter starts with a short historical review of how the Lattice-Boltzmann method emerged from its cellular automata Lattice-Gas roots. A short description of Lattice-Gas methods and a comparison with Lattice-Boltzmann is presented in the second section. The final part of this chapter very briefly describes the derivation of the Lattice-Boltzmann equation from the Boltzmann equation.

2.1 Cellular Automata: A Short History¹

The Lattice-Boltzmann method, as it is currently used in computational fluid dynamics, has its roots in the concept of cellular automata, based on ideas dating back to the 1940s (for details see e.g. [16, 17]).

Inspired by the idea of imitating the behaviour of the human brain and based on suggestions by Ulam [18], the computer pioneer von Neumann developed the concept of a selfreproducing machine to solve highly complex problems. The basic framework suggested by von Neumann consists of a fully discrete universe made up of cells. These cells are characterised by a discrete set of internal states, which are updated in discrete time steps. While the initial and highly complex self-replicating cellular automata suggested by von Neumann [19] is primarily of theoretical interest, a variety of simplified cellular automata for modelling the behaviour of living species were developed in the framework of 'artificial life'.

A particularly popular model suggested in the 1970s is Conway's 'game of life' [20], a two state model residing on a square lattice with cells updating according to the binary state of their four next neighbours. Ten years later, Wolfram classified a family of simple one-dimensional rules (the so called 'Wolfram rules' [21, 22]). He noticed that from a cellular automata with a few simple local rules many features of a complex continuous system can emerge. Thus, the concept of 'complexity' based on mathematical models could be studied by exact computer simulations which (due to their Boolean nature) did not suffer from numerical errors, as it is the case for the more traditional approaches.

¹This section is based on the more detailed historical review given in Chap.1 of the book by Chopard and Droz [10].

Toffoli, Margolus and Fredkin in the mid-1980s recognised cellular automata as a possible model framework for the simulation of physical systems and they developed dedicated hardware (the CAM-6 and CAM-8 machine) for the highly performant simulation of many different cellular automata applications [23].

The first cellular automata fluid flow simulator is based on the so called HPP lattice gas models by Hardy, Pomeau and de Pazzis [24]. Originally developed in the 1970s as a theoretical model for the study of interacting particles, its implementation in the framework of cellular automata raised the question whether or not such a model would be able to simulate the behaviour of a real fluid or gas. Alas, this initial model soon turned out to be insufficient for correctly simulating flow as governed by the Navier-Stokes equation.

With the FHP model published in 1986 by Frisch, Hasslacher and Pomeau [25] and almost simultaneously by Wolfram [26], an accurate lattice gas model for the simulation of fluid flow was available for the first time.

The hope of replacing classical numerical methods in CFD and even wind-tunnels, as for example expressed in an article on the front-page of the Washington Post on November 11, 1985, could not be fulfilled.² Further theoretical developments of the method, mainly driven by the group around d'Humières and Lallemand at the Ecole Normale Supérieure in Paris lead to a variety of improvements, though the major drawback, a relatively high, fixed viscosity and a lack of Galilean invariance of the scheme could not be cured.

By the end of the 1980s, McNamara and Zanetti [27], and Higuera, Jimenez and Succi [28] presented the idea of replacing the Boolean dynamics by calculating the time evolution of a probability density distribution of the particles. This model, commonly referred to as 'Lattice-Boltzmann', turned out to be more suitable for the simulation of a broad variety of real-world flow phenomena.

When the Lattice-Boltzmann method was derived from Lattice-Gas, overcoming the drawbacks of a fixed very high viscosity and a lack of Galilean invariance, first attempts of realistic engineering applications could be made.³ Implementing and running a large-scale three-dimensional Lattice-Boltzmann code to simulate fluid flow inside an industrial device was quite a challenge at this time: the method was instable and not enough was known about how to implement inlet, outlet and particularly wall boundaries. It was the time of 'Legoland'-geometries, where a simple 'marker and cell' approach identified the solid fraction of the flow domain on a regular equidistant lattice. Nevertheless, it was demonstrated in a variety of publications (part of which contributed to this thesis), that the method can be used to produce quantitatively accurate results, which were often difficult or impossible to generate with the same effort using Navier-Stokes or other well established approaches.

A variety of further improvements, particularly the introduction of the BGK single-time relaxation operator [29] and years later the multi-relaxation scheme [30, 31], improved wall boundary conditions [32, 33, 34, 35], local mesh refinement and non-uniform grids [36, 7, 37] allowed the Lattice-Boltzmann method to grow into a mature tool in the area of computational fluid dynamics.

²From today's point of view, it can be said that the limits of the applicability of these early models was not sufficiently understood at that time.

³It was around this time, that the author began his own research within an environment strongly focused on engineering.

However, even today a variety of publications by the community show reluctance to overcome the hardships of applying the method to real world problems by producing quantitatively validated results for academic or engineering problems.

2.2 From Particles to Fluids⁴

For the reader familiar with Navier-Stokes CFD and not with the concept of the Lattice-Boltzmann method, a short review of Lattice-Gas methods and the connection to Lattice-Boltzmann is given in the following.⁵

The underlying principle of Lattice-Gas cellular automata and later the Lattice-Boltzmann method is the numerical simulation of simplified molecular dynamics of the fluid. This is done by evaluating a time and space discrete Boltzmann equation, the so called Lattice-Boltzmann equation [25]. Macroscopic values such as pressure and velocity can be obtained from the fluid density distributions, which (under some assumptions) have a behaviour governed by the Navier-Stokes equation [28, 40].

Before describing the Lattice-Boltzmann method, we first briefly introduce the Lattice-Gas approach in the form of the so called FHP-I automata.

2.2.1 Lattice-Gas (FHP-I)

The FHP automata was first proposed by Frisch, Hasslacher and Pomeau in 1986 [25]. The underlying principle of this approach is that binary 'particles' with unit mass are propagated on a hexagonal Bravais lattice (see Fig. 2.1) in discrete time steps with unit velocity. It should be noted that a triangular or quadrilateral lattice, as commonly used in connection with finite volume or finite element techniques, would cause anisotropic flow.

The particle position and a (discrete) velocity vector are indicated by its lattice coordinates and position at the node respectively. Every node splits into six cells as indicated in Fig. 2.2, showing as an example, a lattice node with particles at cells 1, 2 and 5. An exclusion principle is imposed that prevents the location of more than one particle per cell at the same time. The six possible velocity vectors \vec{c}_i are:

$$\vec{c}_j = [\cos(2j\pi/6), \sin(2j\pi/6)], \ j = 1, ..., 6$$
 (2.1)

A flow diagram of the algorithm is indicated in Fig. 2.3.

The propagation step consists of moving all particles to the next node in the direction of their velocity vectors, with the restriction that propagation is only possible between the six nearest neighbours along the lattice connection lines (see Fig. 2.4).

⁴The research work presented in this section was performed at LSTM Erlangen (Germany) under supervision of M.Schäfer and F.Durst. It was presented at the Journée Gas sur Réseau (Paris, France, 1994) and the ICA Seminar on Modelling and Computation in Environmental Sciences (Stuttgart, Germany, 1995), published in [38, 39]. Financial support by the Bayerische Forschungsstiftung in the Bavarian Consortium of High-Performance Scientific Computing (FORTWIHR II) is gratefully acknowledged.

⁵All details of the theory can be found in a variety of text books, see e.g. [9, 10, 11, 12, 13].



Figure 2.1: Hexagonal Bravais lattice (one lattice node is highlighted).



Figure 2.2: Lattice node with particles at cells 1, 2 and 5.



Figure 2.3: Flow diagram of the cellular automata algorithm.

Depending on the particular model employed (FHP-I, -II or -III), collision rules of varying complexity may change the velocity vectors of those particles entering a collision state after the propagation step. A minimum set of collision rules required to reproduce fluid flow are the head-on and three particle collisions (see Fig. 2.5), which are used in the FHP-I model. The particle mass and momentum is conserved during collisions.

To obtain directed flow (and a pressure gradient), the velocity vectors of a certain amount of randomly chosen particles must be changed at every iteration. To produce flow in the (positive) x direction, for example, particles from cell 3 are moved to cell 6 at the corre-



Figure 2.4: Propagation: particles moving to new positions.



Figure 2.5: Head-on and three particle collisions (the local modification of the post-advection distribution is shown).

sponding nodes as illustrated in Fig. 2.6 (for the numbering of the cells see Fig. 2.2). The resulting local disturbance is propagated through the lattice and, after a certain number of iterations, an equilibrium between acceleration and viscous forces (imposed by collisions and boundary effects, see below) evolves.



Figure 2.6: Acceleration: randomly chosen particles (left image, indicated by dashed circles) change their velocity vector.

In order to obtain macroscopic values, such as pressure and velocity, from the particle distributions, appropriate time and space averaging procedures have to be done. By denoting $n_i(\vec{r}_*, t_*) \in (0, 1)$ as the binary particle density at cell $i \in (1, 2, ..., 6)$ of lattice node \vec{r}_* at time t_* , for example, the mean velocity can be obtained by:

$$< u >= \frac{1}{M} \sum_{t_*=1}^{M} \frac{\sum_i \vec{c}_i \sum_{\vec{r}_*} n_i(\vec{r}_*, t_*)}{\sum_i \sum_{\vec{r}_*} n_i(\vec{r}_*, t_*)}$$
(2.2)

where M is the number of iterations considered for averaging.

2.2.2 Lattice-Boltzmann

Due to the discrete nature of the binary FHP automata, stochastic noise is present which necessitates averaging over long time and large areas. This can be avoided by using particle densities as ensemble averages of the Boolean values. The particle collision rules are replaced by a relaxation operator (last term in Eqn. 2.3) which produces the new particle distributions after the collisions. The Lattice-Boltzmann equation can be written as:⁶

$$f_i(\vec{r}_* + \vec{c}_i, t_* + 1) = f_i(\vec{r}_*, t) + \Omega(f)$$
(2.3)

The densities f_i are equivalent to the ensemble averages of the binary particle densities n_i , where the exact definition of the collision operator $\Omega(f)$ depends on the details of the model employed.

Since no fluctuations are present, the advantage of this approach is that computations can be performed on smaller grids with fewer iterations and the final values can be directly obtained without any time and space averaging processes. This effect is illustrated by a comparison of FHP-I and Lattice-Boltzmann simulations of plane Poiseuille flow, which is shown in Fig. 2.7.⁷ In practical applications, this advantage compensates the less efficient and slower operations (from the computational point of view) with real numbers (particle densities) required for the Lattice-Boltzmann method, compared to the more efficient logical operations on binary particles with an FHP approach.



Figure 2.7: Mean flow rate (y axis) of plane Poiseuille flow vs. iteration number (x axis) for FHP (left) and LBA (right). The dashed lines indicate the theoretical value.

⁶More details of the derivation are presented in section 2.2.5.

⁷The two figures are related to two different simulations with regards to viscosity and Mach-number and one iteration represents a different time-step in each method. The figures are shown to illustrate the statistical noise present in Lattice-Gas and absent in Lattice-Boltzmann.

2.2.3 Fluid-Solid Boundary Conditions

The Lattice-Boltzmann equation 2.3 concerns the time-evolution of a single quantity, the density distribution function f_i . It is therefore natural to treat boundary conditions not in terms of the macroscopic flow variables, pressure and velocity, but through the distribution function itself.

In the case of a fluid-solid boundary, the question is how to reconstruct the distributions which are not updated during the propagation process, because their connecting links reside on the solid surface. The first, and still very popular, approach is the so called standard bounce-back procedure: particles or particle densities which would be moved to occupied nodes are simply bounced back, that is, reflected towards their *incoming* direction. This is resulting in a zero mean velocity at the boundary, located half-way between a boundary fluid node and its adjacent solid node (no slip boundary condition).

In both cases (Lattice-Gas and Lattice-Boltzmann) solid surface boundary conditions can be implemented easily by marking lattice nodes of the solid phase as 'occupied nodes', and applying the bounce-back rule.⁸

The same bounce-back principle is applied to fixed walls at the lattice boundaries.

This procedure allows for the simple implementation of arbitrary complex structures (see, for example, the simulation of flow through a porous sedimentary layer, as presented in Fig. 2.10) or even to change the solid structure during computation, as is necessary for problems with time-varying geometry.⁹

When applying this so called marker-and-cell approach, one has to ensure that the smallest elements of the discretised structure are large enough to prevent a finite-size effect [43, 44, 45], which may lead to unphysical results if the number of non-occupied cells between the obstacles is too small.

Of further consideration is a sufficient resolution of the geometry to capture all relevant details. In case of porous media, certain characteristic quantities such as porosity or the hydraulic radius must not be changed during the voxelisation.

The accuracy of these standard 'bounce-back' boundary conditions is discussed at length in the literature. More complex approaches for reconstructing the missing distributions have been suggested in the mid-1990s (see e.g. [32, 33, 34, 46, 47]). The drawback of all of these 'first-generation' advanced boundary conditions is their suitability for only a reduced set of geometries and a significant amount of additional computational effort required.¹⁰

Conversely, it has been demonstrated (see e.g. [48]) that under appropriate conditions (flow viscosity not too high and large enough lattices), the bulk flow is of second order accuracy in space and time. He et al. [49] showed by theoretical and numerical analyses that the bounce-back scheme is of second order accuracy for plane channel flow. A detailed analysis

⁸Such a mesh, defining 'free' and 'occupied' nodes on an orthogonal regular grid is also termed a 'voxelmesh', and its generation from CAD or tomography data is sometimes called 'voxelisation' (see Chap. 2.2.4).

⁹An example in the framework of medical physics is given in Chap. 5.3 for the simulation of milk/blood clotting. The simulation of coral growth by Kaandorp et al. [41, 42] is an excellent application of Lattice-Boltzmann with flow dependent geometry in marine biology.

¹⁰In case of a sparse implementation as described in Chap. 3.1, the bounce-back condition can be integrated in the preprocessing step, resulting in zero computational overhead.

by Kandhai et al. [50] proves the standard bounce-back method to be of second order accuracy, when the boundary position is assumed to be exactly between the last fluid and the first solid node.¹¹

A clearly unwanted feature of the standard bounce-back method in combination with the BGK relaxation scheme (see Chap. 2.2.5) is a slight dependency of the wall position on the relaxation parameter ω . This error was found to be below 0.1% by Kandhai et al. [50] for a wide range of relaxation parameters. Ferréol and Rothman described this slight relaxation parameter dependency of the wall position already in 1995. They observed that, regardless of its very small quantity, it might have a significant impact on the accuracy of porous media flow simulations, if the mean pore size is very small [51].

A recent study by Pan [52] comparing a variety of advanced Lattice-Boltzmann (TRT or MRT) schemes [30, 31], in combination with the relatively new interpolated boundary conditions [53, 54] and multi-reflection schemes [55], concludes that only the combination of either TRT or MRT with an advanced boundary condition leads to a viscosity independent position of the wall.¹²

In the light of this controversial discussion, we are not convinced that the standard bounceback boundary condition in combination with a BGK relaxation scheme necessarily leads to inaccurate results. However, special care has to be taken to select the relaxation parameter accurately and keep it constant for specific parameter studies. The choice of a more advanced scheme is usually accompanied by a reduced performance, so it might well be that for carefully prepared simulations the standard bounce-back scheme will survive for another decade.

Interpolation schemes appear to be promising candidates for 'state of the art' wall boundary conditions, since they allow a fixed arbitrary position of the wall between two lattice nodes. This leads to increased accuracy of the discretised surface boundary with relatively small computational overhead. Within the preprocessing step, the additional information of the distance between the last fluid node and the solid boundary (the 'q-values') have to be provided (for details see [53]).

For all studies presented in this thesis, the standard bounce-back wall boundary condition was applied. Special care was taken (where required) to ensure a fixed position of the wall by keeping a constant relaxation parameter, or varying it only in a relatively 'safe' low viscosity limit. A sufficiently fine discretisation of the geometry and absence of finite-size effects was verified (where possible and required) by performing mesh convergence studies.

2.2.4 Geometry Discretisation

Due to the relative simplicity of integrating arbitrary complex geometries without impact on the performance, the Lattice-Boltzmann method at first became popular in the area of

¹¹Defining the fluid-solid surface on the boundary nodes itself leads to only first order accuracy.

¹²This statement is in contrast to the author's recent observation on simulating porous media flow with a TRT scheme using standard bounce-back wall boundary conditions. The position of the wall was found to be constant over a wide range of relaxation parameters.

complex geometry flow, where the standard Navier-Stokes based methods caused severe complications for the mesh generation. It soon became obvious that the simple marker and cell approach allows fast and semi-automatic mesh-generation from digital images, CAD-data and other synthetic geometries. The major strategies for geometry discretisation as applied in this thesis are briefly described in the following.

Application Strategy I: Digital Images

For research into porous media flow, the development of new devices in the area of chemical engineering (e.g., heterogeneous catalytic reactions, see Chap. 5.1), or in the area of medical physics concerning patient specific data (see Chap. 6.3 and Chap. 6.4), a detailed knowledge of flow properties inside highly complex geometries is required. It is often not possible to perform mesh generation for such geometries with conventional methods. Using 3D computer tomography (CT), arbitrary complex structures can be digitized and the CT data can be converted into Lattice-Boltzmann voxel data (see Fig. 2.8).¹³



Figure 2.8: Flow diagram illustrating the voxel-mesh generation from real objects.

To illustrate the capabilities of the method, the fluid flow through a digitised electron microscope picture of a sedimentary layer taken from the north German sea shore as shown in Fig. 2.9 was simulated.¹⁴

The computation, originally performed in the year 1993 on a Convex C-210, was repeated in 1994 on one processor of a Cray-YMP. 10,000 iterations were required for a 500×500 lattice to reach the steady state. This corresponds to 1200 seconds of total CPU time,¹⁵

¹³Sometimes, as in the case of medical data, a more or less complex reconstruction procedure has to be performed in order to identify the exact position of solid walls.

¹⁴This approach was developed by the author during his time as a diploma student at the Institute of Biology and Chemistry of the Sea (ICBM) at the University Oldenburg (Germany) under the supervision of Prof. Schellnhuber, without being aware of the work simultaneously carried out by Ferréol and Rothman on Lattice-Boltzmann simulations of flow through Fontainebleau sandstone [51], the first publication on flow through real digitised geometries with the Lattice-Boltzmann method.

¹⁵With a state of the art implementation it would take less than 10 seconds to produce the same result on one CPU of an NEC SX-8 vector computer.



Figure 2.9: Digitised electron microscope picture of a sedimentary layer, from [56].

where $1.47 * 10^6$ lattice updates per second were accomplished with a performance of 190 million floating point operations per second (MFLOP/s) from 330 MFLOP/s theoretical peak performance.¹⁶ This indicates the possibility of a straightforward vectorisation of the Lattice-Boltzmann approach, which constitutes a great advantage of this method.

Fig. 2.10 shows the computed velocity vectors for the sediment layer. Constant velocity inlet and pressure outlet boundary conditions where applied, at the upper and lower boarder of the domain periodicity was considered. This illustrate that, regardless of the complexity of the pores, the flow features are well captured.

Application Strategy II: CAD-data

In modern car design, a complete model of the automobile is available as CAD data. Using dedicated software, the geometry description for the Lattice-Boltzmann simulation in terms of voxel-meshes can be generated almost automatically from the CAD data (see Fig. 2.11).

As an illustration, the turbulent flow simulation around two geometries defined in terms of CAD data are shown in Fig. 2.12.

Application Strategy III: Synthetic Geometries

Occasionally, neither a real geometry nor CAD data are available as a description of the solid fraction. Particularly for academic flow studies, generic shapes consisting of regularly or randomly positioned rectangles or boxes [44, 39], packed beds of spheres or randomly

¹⁶Today's figures for a performance optimised 3D code are 40 million lattice site updates per second with a performance of 6 GFLOPS on an NEC SX-8 machine with a peak performance of 10 GFLOP/s. For further performance considerations see Chap. 3.1.



Figure 2.10: 2D Lattice-Boltzmann simulation of flow in a sedimentary layer (every tenth velocity vector in both dimensions is plotted). Geometry courtesy of J. Kropp, University of Oldenburg, Germany. Flow-solver: LB-FLOW, University of Oldenburg (1993), hardware: Convex C-210.



Figure 2.11: Flow diagram illustrating the voxel-mesh generation from CAD data.

arranged spheres [57, 58, 52] can be considered. The procedure of generating the voxel mesh for these synthetic geometries is described in the following flow-chart Fig. 2.13.

As an illustration of this approach, the flow through a complex synthetic geometry generated by a computer simulation of a spinodal decomposition process is shown in Fig. 2.14.



Figure 2.12: Left: Streamlines around a motor engine. Right: Turbulent flow field around an ASMO shape. Geometry data courtesy of INVENT Computing GmbH, Germany. Flow-solver BEST, LSTM Erlangen (1998), hardware: Cray Y-MP.



Figure 2.13: Flow diagram illustrating the voxel-mesh generation from synthetic geometries.

The above discretisation approaches become more demanding if more complex computational grids or boundary conditions are considered. For extrapolated boundary schemes which take into account the exact position of the wall between two lattice nodes, the qvalues have to be computed (see Chap. 2.2.3).

As a minimum requirement for local mesh refinement [7, 59, 60, 61], the option to define refined areas must exist. In the case of octree-based refinement (see e.g. [62, 63]), significant effort is required in the preprocessing step to generate an adequate mesh.

Other more exotic approaches for Lattice-Boltzmann simulations on irregular meshes are described in the literature (see e.g. [37, 64, 65, 36]). Alas, the additional computational effort to extrapolate the distributions (a simple propagation step as in the case of regular equidistant meshes is no longer possible) is too high to make Lattice-Boltzmann a competitive tool for practical applications on these types of meshes.



Figure 2.14: Flow through the solid phase created by spinodal decomposition. An isosurface of the velocity is shaded by the pressure. Geometry data courtesy of Mitsui Chemicals, Japan. Flow-solver: MUSUBI, CCRLE, NEC Europe Ltd. (2002), hardware: NEC SX-4.

Following this introduction to the method, the Lattice-Boltzmann theory will be briefly reviewed in more detail in the following section.

2.2.5 From the Boltzmann Equation to Lattice-Boltzmann¹⁷

From a pragmatic point of view, it is sufficient to show that the Navier-Stokes equation (which describe the mass and momentum conservation of the flow) can be derived from the Lattice-Boltzmann equation. It is interesting to remark that this follows the historical development of the method from its Lattice-Gas origins, based on the observation that a variety of different non-Physical 'micro worlds' can result in the same physically correct 'macro world'. It is sufficient to ensure certain symmetries in the microscopic rules (translation-invariance, rotation-invariance (isotropy), parity invariance and Galilean invariance), to achieve the correct number of conserved quantities of the macro system.

Eight years after the first publications on Lattice-Boltzmann, it was demonstrated how the Lattice-Boltzmann equation can be derived from the Boltzmann equation itself [66, 67], which is, from a theoretical point of view, a slightly more satisfying approach. Since it is well established, how and under which conditions the Navier-Stokes equation follows from the Boltzmann equation, for the theoretical justification of the Lattice-Boltzmann method it is sufficient to derive the equations to be applied for a numerical simulation directly from the Boltzmann equation.

¹⁷This section follows a description given by Krafczyk [8].

In the following we will very briefly sketch this derivation (for details, the reader is referred to the literature and textbooks [9, 10, 11, 12, 13]). For simplicity, we restrict ourselves to the simple BGK-approximation of the collision term.

The probability distribution of the particles at the time *t* located at \vec{x} with velocity $\vec{\xi}$ can be described in the framework of Statistical Physics by the Boltzmann-equation (Eqn. 2.4).

$$\frac{\partial f}{\partial t} + \vec{\xi} \cdot \frac{\partial f}{\partial \vec{x}} + \vec{F} \frac{\partial f}{\partial \vec{\xi}} = \Omega(f)$$
(2.4)

The time development of the distribution function f is determined by the advective term (second term of the left-hand side), an external force \vec{F} (third term on the left-hand side) and the collision operator (right-hand side). The collision operator $\Omega(f)$ models the interaction of particles involved in a collision.

From Statistical Physics it is known that a velocity distribution of a homogeneous gas in equilibrium can be described by the Maxwell-distribution (Eqn. 2.5):

$$f^{eq} = \frac{\rho}{(2\pi c_s^2)^{d/2}} \exp\left(-\frac{(\vec{\xi} - \vec{u})^2}{2c_s^2}\right)$$
(2.5)

where \vec{u} is the average velocity of the particles, c_s the speed of sound, and d the dimension of space. This distribution function is also referred to as 'local equilibrium'.

Bhatnagar, Gross and Krook [68] suggested a simplification of the collision operator (right term of Eqn. 2.4) by the assumption that collision results in a local relaxation towards the Maxwellian equilibrium (Eqn. 2.5). Therefore, the distribution function f locally relaxes towards its equilibrium distribution function f^{eq} .

Assuming an average relaxation time τ , related to the average time between two particlecollisions, the collision term of Eqn. 2.4 can be replaced by a one-step relaxation process:

$$\Omega = -\frac{1}{\tau} \left(f - f^{eq} \right) \tag{2.6}$$

Using this relation, the Boltzmann-equation (Eqn. 2.4) can be rewritten as:

$$\frac{\partial f}{\partial t} + \vec{\xi} \cdot \frac{\partial f}{\partial \vec{x}} + \vec{F} \frac{\partial f}{\partial \vec{\xi}} = -\frac{1}{\tau} (f - f^{eq})$$
(2.7)

A solution of the Boltzmann-equation in this form is usually not practical. With the assumption that the considered fluid is a dense gas and the deviation of f from its local equilibrium is f^{eq} is small, a discretisation of the velocity-space with just a few degrees of freedom is possible. The continuous distribution f is now replaced by its velocity-discrete equivalent f_i , which propagates with the discrete velocity \vec{e}_i .

The Boltzmann-equation (Eqn. 2.7) changes (neglecting external forces here) into the discrete Boltzmann-equation (Eqn. 2.8) as follows:

$$\frac{\partial f_i}{\partial t} + e_{i\alpha} \frac{\partial f_i}{\partial x_{\alpha}} + -\frac{1}{\tau} \left(f_i - f_i^{eq} \right)$$
(2.8)

where i = 1, ..., N enumerate the discrete velocities (degrees of freedom) \vec{e}_i with indices α describing the components of the Cartesian coordinates.

For the purpose of numerical simulation, the velocity-discrete Boltzmann-equation (Eqn. 2.8) is discretised in space and time by a finite-difference scheme. With $c_s = 1/\sqrt{3}c$ and applying a lattice described by its lattice-vectors $\vec{e}_i = \frac{1}{c}\vec{e}_i$, Eqn. 2.8 modifies into:

$$\frac{f_i(t+\Delta t,\vec{x}) - f_i(t,\vec{x})}{\Delta t} + c \frac{f_i(t+\Delta t,\vec{x}+\vec{e}_i\Delta x) - f_i(t+\Delta t,\vec{x})}{\Delta x} = -\frac{1}{\tau} \left(f_i(t,\vec{x}) - f_i^{eq}(t,\vec{x}) \right)$$
(2.9)

With a space-discretisation $\Delta x = c \Delta t$ and multiplication with Δt the Lattice-Boltzmann equation can be derived:

$$f_i(t + \Delta t, \vec{x} + \vec{e}_i \Delta x) = f_i(t, \vec{x}) - \frac{\Delta t}{\tau} \left(f_i(t, \vec{x}) - f_i^{eq}(t, \vec{x}) \right)$$
(2.10)

With an appropriate choice of the lattice and equilibrium distribution f^{eq} , this equation can be used for the numerical simulation of the time-evolution of the distribution function f.

For the three-dimensional case, lattices with sufficient symmetry are typically the so called D3Q15 or D3Q19 models (see Fig. 2.15 for the more common D3Q19 lattice).



Figure 2.15: Stencil of a D3Q19 lattice.

In the limit of small Mach numbers, a Taylor-expansion of the Maxwellian velocity distribution (Eqn. 2.5) leads to the equilibrium distribution (Eqn. 2.11).

$$f_i^{eq} = t_p \rho \left\{ 1 + \frac{e_{i\alpha}u_{\alpha}}{c_s^2} + \frac{u_{\alpha}u_{\beta}}{2c_s^2} \left(\frac{e_{i\alpha}e_{i\beta}}{c_s^2} - \delta_{\alpha\beta} \right) \right\}$$
(2.11)

with t_p as a direction-dependent weighting factor which takes into account the appropriate contribution of the links (orthogonal: p = 1, diagonal: p = 2 and center: p = 0).

For the D3Q19 model, which was used throughout this work, $t_0 = 1/3$, $t_1 = 1/18$, $t_2 = 1/36$. The quantity c_s is the (also model-dependent) speed of sound (for D3Q19: $c_s = 1/\sqrt{3}$).

Below it is briefly explained how the macroscopic quantities present in the Navier-Stokes equation (pressure and flow velocity) are computed from the density distribution f in the Lattice-Boltzmann equation (Eqn. 2.10).

The fluid density ρ and the velocity components u_{α} can be derived in terms of the zeroth and first moment of the density distribution function:

$$\rho = \sum_{i} f_i = \sum_{i} f_i^{eq} \tag{2.12}$$

$$\rho \vec{u} = \sum_{i} \vec{e}_i f_i = \sum_{i} \vec{e}_i f_i^{eq}$$
(2.13)

The pressure *p* is given by an equation of state as

$$p = \rho c_s^2 = \frac{1}{3}\rho$$
 (2.14)

and the viscosity is related to the relaxation time τ (the inverse of the particle collision frequency ω) by

$$v = \left(\frac{\tau}{3} - \frac{1}{6}\right) \tag{2.15}$$

and the stress-tensor is given by

$$S_{\alpha\beta} = -\left(1 - \frac{1}{2\tau}\right) \sum_{i} e_{i\alpha} e_{i\beta} \left(f_i - f_i^{eq}\right)$$
(2.16)

In the limit of small Mach and Knudsen numbers, these quantities are a solution of the Navier-Stokes equation, which can be written as:

$$\rho\left(\frac{\partial \vec{u}}{\partial t} + \vec{u}\,\nabla u\right) = -\nabla p + \nabla S \tag{2.17}$$

where $\hat{S} = (S)_{\alpha\beta=1,2,3}$ is the strain tensor, defined as

$$S_{\alpha\beta} = \mu \left(\frac{\partial u_{\alpha}}{\partial x_{\beta}} + \frac{\partial u_{\beta}}{\partial x_{\alpha}} \right)$$
(2.18)

with the dynamic viscosity $\mu = \nu \rho$.

It is worth noting that, opposite to methods based on the discretisation of the Navier-Stokes equation, the stress-tensor (Eqn. 2.16) is a local quantity within the Lattice-Boltzmann framework. It is computed from the non-equilibrium part of the density distribution and
does not require a (possibly more complicated) derivation from the velocity field. This feature is particularly used for the implementation of subgrid turbulence models (see e.g. [69, 70, 71, 72, 73]) or non-Newtonian flow (see Chap. 6.3.3), where the viscosity is a shear dependent local quantity.

Chapter 3

Implementation Challenges

In this chapter, some aspects of a performance oriented implementation are presented. A three-dimensional simulation involving several ten million fluid nodes on a PC-cluster or vector-supercomputer can currently be considered as state of the art. Depending on the solid fraction, the bounding box might contain one order of magnitude more lattice nodes (fluid plus solid) in total.

The Lattice-Boltzmann method is often said to be very efficient and easy to implement. But, in most cases described in the literature, a simple full matrix implementation is used, a method where not only the fluid nodes but also the solid fraction is allocated in the computer's memory.

The difference between a straight-forward coding of the basic equations and a sophisticated (often hardware optimised) implementation can easily make a difference of one order in memory consumption and up to two orders in CPU-time. Therefore, a non-optimised simple implementation, solving one million grid points can require ten days, whereas a ten times larger optimised implementation will finish on the same machine within 24 hours.

Moreover, special hardware such as parallel computers or vector architectures require specific implementation techniques, otherwise the code would run extremely slowly, if at all. Conversely, general purpose cache-based machines benefit from a hardware-optimised layout of the data in the computer's memory, considering the cache-size, number of cache lines etc. [74].¹

¹These details are not presented here, since they will be subject of the PhD thesis of Thomas Zeiser, a researcher working in close co-operation with the author.

3.1 Performance Optimised Implementation²

The following section is focused on the performance optimised implementation primarily on vector architectures. The sparse implementation approach described below has been well established in linear algebra for several decades, but has not been applied to the Lattice-Boltzmann method until recently [79, 75].

In the following, the advantage of this method in comparison to the still widely used full matrix implementation is demonstrated.

3.1.1 Full Matrix versus Sparse Implementation

In the framework of a simple full matrix implementation, the density distribution array for all nodes (solid and fluid) within the bounding box is allocated in the computer's memory. This results in (2*)19*lx*ly*lz REAL numbers for the D3Q19 model for an lx*ly*lz lattice.

Sparse implementation techniques were first applied to the Lattice-Boltzmann method by Schulz et al. [79], suggesting storage of the density distribution only for the fluid nodes. This requires keeping an adjacency list for the next neighbours' addresses, but (depending on the geometry) can save considerable memory. Only (2*)N*19 REAL numbers for the density distribution (N=number of fluid cells) and N*19 INTEGERs for the adjacency list have to be stored in the case of a sparse Lattice-Boltzmann implementation.

The adjacency list is required for looking up the addresses of the next neighbouring lattice nodes during the advection step. This address look-up is not necessary in the full matrix approach, since the neighbourhood can be determined by easy index algebra, making use of the fact that the geometric topology is mapped 1:1 onto the three-dimensional arrays storing the distributions.

In the case of a sparse implementation, the voxel mesh defining the solid fraction in terms of Cartesian coordinates of the occupied lattice nodes has to be preprocessed: the fluid fraction is then mapped onto a graph, giving a unique identifier for each node and defining its connectivity (see Fig. 3.1).

Depending on the hardware considered for the simulation, one might prefer to either store all distributions belonging to one node in a contiguous array (array-of-structures layout), or first store all densities of direction 0, then all densities of direction 1 etc. (structure-of-arrays layout).³

Another performance relevant rearrangement of the data-layout can be achieved by 'blocking' adjacent groups of lattice nodes in a suitable way that they fit into the computer's cache

³Hardware dependent details in the performance of these two approaches are discussed in [74, 80].

²The research work presented in this section was performed at the C&C Research Laboratories, NEC Europe Ltd. (St.Augustin, Germany). It was presented at the Second International Conference for Mesoscopic Methods in Engineering and Science - ICMMES 2005 (Hong Kong, China, 2005 - invited talk), the 4th Teraflop Workshop (Stuttgart, Germany, 2006 - invited talk), the Third International Conference for Mesoscopic Methods in Engineering and Science - ICMMES 2006 (Hampton, Virginia/USA, 2006 - invited talk) and at the 15th Discrete Simulation of Fluid Dynamics conference - DSFD 2006 (Geneva, Switzerland, 2006), published in [75, 76, 77] and submitted for publication in [78].



Figure 3.1: Mapping of fluid nodes onto a graph and construction of the adjacency list.

(see e.g. [80]). Thus, several iterations can be performed in the cache, making use of its faster memory access.

It is notable that such flexibility of arranging an optimal data-layout is not possible within the framework of a full matrix implementation. Alas, it will also be difficult to control these details when the implementation is carried out in modern object-oriented programming languages. It remains to be shown that the trade-off of an elegant C++ implementation is not the loss of at least one order of magnitude in performance.⁴

Although the above indicates some obvious advantages of the sparse method compared to the full matrix implementation, it is necessary to see if the required indirect address lookup causes a computational overhead and delay in memory access, which maybe outweighs all advantages. Therefore, a direct comparison of a lattice-BGK full matrix and sparse implementation was undertaken.

3.1.2 Performance Measurement on a Vector Computer

When estimating the efficiency of a full matrix versus a sparse implementation, three figures are of interest:

- *MFLOP/s* (million floating point operations per second): in comparison to the theoretical peak performance, this figure indicates how efficient the implementation is for the given hardware. A good MFLOP rate relates to an optimal use of the CPU with few idle cycles while waiting to load or store data. This figure is usually not related to the wall clock time for solving a given problem, since particularly a non-optimised implementation of computationally demanding equations can lead to high MFLOP/s.
- *MLUP/s* (million lattice site updates per second): the MLUP/s gives the update rate of the code and thus the total speed of the implementation.⁵ A high MFLOP/s number does not necessarily result in high MLUP/s, since a more efficient implementation of

⁴The author is extremely keen to see benchmark results produced with a high performance C++ Lattice-Boltzmann implementation.

⁵For estimating the MLUP/s as a measure for the actual speed to solve a given problem, only the *fluid nodes* have to be considered. Counting all (fluid and solid) lattice nodes within the bounding box would

the equations to be solved (requiring less floating point operations per cycle) could reduce the MFLOP/s while actually increasing the MLUP/s. The MLUP rate directly relates to the wall clock time for solving the problem.

• *MBYTE*: the total memory required to store the density distribution array (and the adjacency list for the sparse code) shows which implementation strategy - sparse or full matrix - is more efficient with regard to memory. Although we expect to require some additional memory for the adjacency list, we save memory from a certain fraction of occupied lattice nodes, since only memory for the fluid nodes has to be allocated.

Geometries

To estimate the performance with respect to the aforementioned quantities, a set of twelve different geometries was considered; from an empty square channel over porous media to medical geometries such as an aorta and a cerebral aneurysm (see Fig. 3.2).

With a great variety of porosity, specific surface and complexity, these geometries represent the most typical problem configurations (for details see Tab. 3.1).

Sample	Bounding Box	Nodes	Fluid Nodes	Porosity
1	100 * 100 * 100	1,000,000	1,000,000	1.0
2	100 * 96 * 96	921,600	454,217	0.49
3	44 * 147 * 147	950,796	677,186	0.71
4	780 * 122 * 122	11,609,520	6,405,404	0.55
5	500 * 80 * 80	3,200,000	2,267,308	0.71
6	64 * 32 * 32	65,536	46,256	0.71
7	128 * 64 * 64	524,288	316,272	0.60
8	256 * 128 * 128	4,194,304	2,530,176	0.60
9	256 * 128 * 128	4,194,304	2,277,168	0.54
10	256 * 128 * 128	4,194,304	2,139,046	0,51
11	263 * 175 * 74	34,058,503	171,166	0,05
12	459 * 121 * 154	8,553,006	494,684	0,06

Table 3.1: Dimensions of the 12 benchmark cases.

Performance Results

All performance results were obtained on CCRLE's NEC SX-6i vector-computer which has a peak performance of 8 GFLOP/s. A summary of all results discussed in the following can be found in Tab. 3.2 at the end of this section. The data in the three columns of the table Tab. 3.2, 'Performance', 'Speed' and 'Memory' are also illustrated in the Figs. 3.3, 3.4 and 3.5 respectively.

naturally always result in a higher rate for the full matrix implementation, but represent a meaningless figure, since the update of solid nodes does not contribute to the solution of the problem.



Figure 3.2: Benchmark cases 2-12 (increasing numbers from upper left to lower right), case 1 is an empty square channel (not displayed here).

MFLOP/s Fig. 3.3 illustrating the first column of Tab. 3.2 shows that for all 12 samples, a performance of approximately 4 GFLOP/s (50 % of the peak performance) is achieved with no strong preference for either method.⁶ This indicates that the full matrix and sparse code can be implemented with equivalent performance on a vector computer for a variety of different geometries. Moreover, it is worth noting that for the samples considered here the performance is roughly independent of parameters such as mesh size or fluid/solid fraction.⁷ This supports the statement of an efficient handling of arbitrarily complex geometries with the Lattice-Boltzmann method.

MLUP/s A significant performance gap between the full matrix and sparse implementation is observed with regard to the fluid lattice site updates per second. Except for the trivial case of an empty square box, all MLUP/s of the sparse implementation are far above the full matrix case (see Fig. 3.4 illustrating the second column of Tab. 3.2). This effect is

⁶The MFLOP/s were measured using the performance analysis software 'ftrace'.

⁷Although it can be assumed that these parameters are responsible for the small variations in the attained performance, due to varying memory access giving preference to the one or other method from sample to sample.

Sample	Performance		Speed		Memory	
	(MFLOP/s)		(MLUP/s)		(MBYTE)	
	sparse	full	sparse	full	sparse	full
1	4555	5444	27.48	31.73	432	268
2	4173	3946	25.17	11.31	224	352
3	4208	4324	25.19	17.57	304	368
4	4367	4188	26.56	13.90	2544	3840
5	4618	4742	28.08	19.83	912	1104
6	3980	3368	23.77	12.58	64	80
7	4171	3852	25.09	13.32	176	224
8	4284	3760	25.94	13.55	1024	1424
9	4329	4484	26.19	14.55	944	1424
10	4366	5050	26.40	15.40	880	1424
11	4253	5207	26.40	1.55	144	1184
12	4338	5404	26.50	1.88	366	2880

Table 3.2: Comparison of performance, speed and memory-consumption between full matrix and sparse implementation for the geometries 1-12 (see Fig. 3.2). The values contained in the three columns are illustrated in the Figs. 3.3, 3.4 and 3.5.



Figure 3.3: Illustration of column 1 of Tab. 3.2: Comparison of full matrix and sparse Lattice-Boltzmann performance (MFLOP/s) for the geometries 1-12 (see Fig. 3.2).

strongest for the medical geometries 11 and 12 (aorta and cerebral aneurysm) with complex thin channels inside a large bounding box, where only a few percent of the domain are fluid nodes. The performance of the full matrix implementation is below 2 MLUP/s, whilst almost geometry-independent, the sparse implementation shows an update rate of approximately 25 MLUP/s.

The reason for the strong variation in the fluid MLUP/s for the full matrix code has its origin in the the specific treatment of if-statements by a vector computer. Although the computational expensive relaxation subroutine is assumed to be executed only on the fluid

nodes (excluding the solid nodes by an if-statement), typically, vector machines compute the relaxation for all nodes and later discards the results for the fraction of if cases which were not fulfilled (so called 'masking'). While the total (solid and fluid) MLUP/s is more or less a constant also for the full matrix code, the strong variations of the fluid MLUP/s reflect the amount of the solid fraction contained in the bounding box.

The little additional cost of the indirect address look-up can be seen in case 1, the square channel. The speed of the full matrix implementation using simple index algebra to find the next neighbour cells for the advection step is only $\approx 10\%$ above the sparse code.



Figure 3.4: Illustration of column 2 of Tab. 3.2: Comparison of full matrix and sparse Lattice-Boltzmann speed (MLUP/s) for the geometries 1-12 (see Fig. 3.2).

MBYTE The computer memory required for both methods (shown in the third column of Tab. 3.2 and Fig.3.5) is obviously dependent on the problem size, hence large variations can be observed amongst the 12 samples in Fig. 3.5.

Except for the free channel (case 1), the memory consumption of the sparse implementation is below that of the full matrix code. The memory reduction, resulting from only allocating fluid nodes, outweighs the cost of storing the adjacency list for all relevant cases. Particularly relevant is the memory reduction to almost 10% for the two medical cases 11 and 12. Opposite to the observation made for the speed measured in MLUP/s, the memory reduction is not an effect specific for vector computers.

3.1.3 Detailed Analysis: The Medical Case

A more detailed analysis for a medical geometry was undertaken for the case 11 (Fig. 3.6). The size of the bounding box is 263 * 175 * 74 = 3,405,850 nodes, of which only 171,166 nodes (5%) are fluid. The figures for both the full matrix and sparse code are listed in Tab. 3.3.



Figure 3.5: Illustration of column 3 of Tab. 3.2: Comparison of full matrix and sparse Lattice-Boltzmann memory consumption (MBYTE) for the geometries 1-12 (see Fig. 3.2).



Figure 3.6: Geometry case 11: medical image showing an abdominal aorta (left trunk) with an aneurysm in the iliac region.

As in the above example, medical geometry data often consist of complex tortuous flow channels, resulting in a very small portion of fluid nodes inside a large bounding box. Compared to a full matrix code, the memory consumption for the sparse implementation is almost an order of magnitude less (12.2 % for case 11, 12.7 % for case 12).

Although the performance of the sparse implementation is below that of a full matrix code (4253 MFLOP/s (sparse) compared to 5207 MFLOP/s (full matrix)), the speed (measured in MLUP/s for the fluid nodes), and such the total time to solve the problem, is more than one order of magnitude better for the sparse code: 26.4 MLUP/s (sparse) require 6.6s for

	Full matrix	Sparse	Comparison (%)
Nodes	3,405,850	171,166	5.0
Memory (MBYTE)	1184	144	12,2
MLUP/s	1.55	26.40	1703
CPU seconds	110	6.6	6.0
MFLOP/s	5207	4253	81.2

Table 3.3: Comparison of sparse and full matrix performance for a medical geometry (case 11).

1000 iterations, while the very low 1.55 MLUP/s (full matrix) need 110s to accomplish the same task.

The enormous gain in performance and reduction of memory for this type of geometries when using a sparse implementation is obvious, the same case can be computed with 5-10% of the resources required to run a full matrix implementation.

These measurements and considerations were the basis on which the decisions for the data layout of the new Lattice-Boltzmann flow solver developed within the *International Lattice-Boltzmann Software Development Consortium* were made.⁸

3.2 MPI-Parallelisation⁹

The concept of parallelisation is distributing the computational effort onto several CPUs, with each CPU working simultaneously on solving a fraction of the problem. Often, this is done by splitting the computational domain into pieces ('partitioning') to be solved individually by each CPU. Usually, this approach requires exchange ('communication') of a fraction of the data (residing in the disjoint address space of each CPU) across each partition's boundary at defined time-steps.¹⁰

This implies, that optimising the partitioning of the computational domain requires consideration of three associated features: which portion of data and which computational tasks

⁸The current version, which was used for the studies in Chap. 5.3 and Chap. 6.3, is amongst the fastest Lattice-Boltzmann flow solvers worldwide.

⁹The research work presented in this section was performed at the C&C Research Laboratories, NEC Europe Ltd. (St.Augustin, Germany) in co-operation with members of the *International Lattice-Boltzmann Software Development Consortium*. An extension of the approach described here was presented as an invited talk at the Second International Conference for Mesoscopic Methods in Engineering and Science - ICMMES 2005 (Hong Kong, China, 2005), the 4th Teraflop Workshop (Stuttgart, Germany, 2006), the Third International Conference for Mesoscopic Methods in Engineering and Science - ICMMES 2006 (Hampton, Virginia/USA, 2006) and as a contributed talk at the 15th Discrete Simulation of Fluid Dynamics conference - DSFD 2006 (Geneva, Switzerland, 2006), published in [75, 76, 77] and submitted for publication in [78]. Since it will also be part of the PhD thesis of L.Axner at the University of Amsterdam (UVA) [81], only a short introduction to the underlying concept is given, as far as it was worked out by the author of this thesis. Financial support of the European projects @neurIST (contract no. IST-027703) and COAST (contract no. 033664) is gratefully acknowledged.

¹⁰For shared-memory systems, the explicit coding of the communication is not required, since all CPUs share the same address space. The distribution of the computational work to the CPUs is handled via extensions of the operating system, using specific compiler directives (such as OpenMP), e.g. to carry out a loop-based parallelisation. This approach is generally easier to implement than an MPI parallelisation.

(these are not necessarily the same) to distribute on which node, and identifying the data involved in communication.

The optimisation is performed under the paradigms of good 'load balancing' and minimal 'communication overhead':

- Memory load balancing: an equal amount of memory should be allocated for each partition.
- CPU load balancing: each partition should require the same amount of CPU-time between the communication.
- Reducing the communication overhead: the amount of data exchange between the nodes must be minimised.

In this chapter, some aspects of a specific approach parallelising a Lattice-Boltzmann implementation are discussed. For motivating the METIS-based partitioning [82] of the previously described sparse implementation, the disadvantages of a simple regular partitioning (which is still frequently applied) are pointed out.

3.2.1 Partitioning

In the literature, it is often stated that Lattice-Boltzmann is an inherently parallel scheme which can be easily parallelised [83, 79, 84] using communication libraries (e.g. implementations of the MPI standard [85]).

Since only next neighbour interactions are considered for the streaming operator within one time-step, this statement is partially true. The absence of complicated or long-range interactions makes it easy to identify the data involved in the communication process: they are simply the outgoing and incoming density distributions at the domain boundaries.

As will become clear in the following section, a regular partitioning scheme can not simultaneously fulfill the three conditions of a successful parallelisation, memory-balance, CPU-load-balance and minimising the communication overhead, as defined above.

Starting with a complex geometry, e.g. a porous media or medical geometry, enclosed by a bounding-box of size lx * ly * lz, simple partitioning is achieved by dividing one or more axis regularly into n $(n_x, n_y \text{ and } n_z)$ pieces.

Each partition contains a subset $n_x * n_y * n_z$ of the original domain, and communication with the neighbouring partitions has to be done via the surfaces at $x = 1, x = n_x, ...$, which is straightforward and easy to implement.

The drawback of this simple approach is obvious, when the geometry considered is not homogeneous: for example when a longer inlet and outlet region (containing no solid fraction) is present, or the solid fraction is irregularly distributed, so some partitions contain a larger amount of fluid nodes than others.

The CPU-time required for completing one iteration in each partition is a function of the number of fluid nodes and not of the size of the bounding box. So it is obvious, that for

inhomogeneous distributions of the solid fraction regular partitioning causes a CPU load-imbalance.

An improvement can be achieved by dividing the computational domain into boxes of various size, each containing the same number of fluid nodes. Obviously, this approach only makes sense for a sparse implementation where only fluid nodes are allocated, otherwise the different sizes of the bounding boxes will cause a memory imbalance.

These considerations confirm that an equal partitioning with regard to memory and CPU consumption can only be achieved in the framework of a sparse implementation. The statement of an 'easy and straightforward parallelisation' appears to be highly questionable.

Even an irregular decomposition of a sparse Lattice-Boltzmann code into different sized domains does not yet take into account our third criterion: minimising and balancing the communication overhead. It is obvious that by a restriction to squared boxes, the number of links connecting to a neighbouring partition can be quite high and differ from partition to partition, depending on the number of fluid nodes contained within the cutting plane separating two adjacent partitions.

Giving up the restriction of square shapes for the partitions allows for minimisation of the number of links involved in the communication process whilst the number of fluid nodes per partition are balanced.

In the context of a sparse implementation the fluid domain is mapped on a graph with a known number of nodes and edges, therefore an optimised decomposition can be considered after weights to the nodes and edges are defined.¹¹ A collection of 'graph-partitioning' algorithms making use of multilevel recursive-bisection and multilevel k-way schemes based on Kernighan-Lin and modified Fiduccia-Mattheyses partitioning algorithms is (for example) provided by the METIS library [82].

The preprocessing step, in which a graph is built up from a certain voxel mesh, is followed by a partitioning step, which distributes the fluid nodes amongst a given number of partitions. The resulting partition can have complex shapes (see Fig. 3.7).¹²

In the framework of a sparse Lattice-Boltzmann implementation, lists of the nodes involved in the communication process (exposed nodes) must be identified within each partition. It has to be known exactly which outgoing link is sending to- and which incoming link is receiving from which neighbouring partition (see Fig. 3.8).

The parallel Lattice-Boltzmann algorithm has an additional communication step following relaxation and propagation. After propagation, the send-buffer (ghost cells, see Fig. 3.9) are filled up by the outgoing densities of the exposed nodes. These outgoing densities are then sent to the neighbouring partitions, where they are stored in the receive-buffer. From the receiver-buffer, the missing incoming links are copied to the exposed cells (see Fig. 3.10).

The computational sub domain of each partition is therefore divided into different types of nodes (see Fig. 3.11):

¹¹The weights are equally distributed if no local mesh refinement or other inhomogeneous processes are used.

¹²The implementation of the partitioner was done by L.Axner (UVA) in close collaboration with G.Berti (CCRLE) and the author.



Figure 3.7: Partitioning of two complex geometries for a sparse Lattice-Boltzmann code by METIS: aorta (left) and catalyst (right). Images created by Lilit Axner (UvA) and taken from [86, 78].



Figure 3.8: Three partitions of a computational domain mapped on a graph. The outgoing and incoming links from neighbouring partitions are identified.



Figure 3.9: Three partitions of a computational domain mapped on a graph. The outgoing links are copied into a ghost-layer during the propagation step.

- Inner Nodes: these nodes are not involved in any communication process.
- *Exposed Nodes:* these nodes are sending to and receiving from neighbouring partitions.



Figure 3.10: Three partitions of a computational domain mapped on a graph. The incoming links are copied from the receive buffer.

- *Ghost Cells:* the outgoing densities are shifted to these links from the exposed nodes during the propagation step. They build up the send-buffer.
- *Receive-buffer:* incoming links of the exposed nodes receiving densities from neighbouring partitions.



Figure 3.11: Three partitions of a computational domain mapped on a graph. Inner nodes, exposed nodes and the receive-buffer are identified.

In principle, it is possible to perform the relaxation-advection step of the inner nodes simultaneously to the communication step of the exposed nodes, thus completely hiding the communication overhead (see Fig. 3.12).¹³

¹³This 'latency-hiding' is not yet implemented into the code considered within this chapter. Preliminary results from a more detailed study [78] indicate that the most significant performance loss is due to load imbalance, and not caused by communication.



Figure 3.12: Flow-chart of an MPI-parallelised Lattice-Boltzmann algorithm with overlapping computation at inner nodes and communication at the boundaries.

3.2.2 MPI Performance Results¹⁴

The sparse Lattice-Boltzmann software developed in the framework of the *International Lattice-Boltzmann Software Development Consortium* has been parallelised as described above, and a performance analysis on a METIS-based partitioning was obtained for the two sample geometries shown in Fig. 3.7 and a free square channel for up to 128 CPUs. The first example consists of a medical geometry (aorta) which was discretised with 5,775,552 fluid voxel; the second example from chemical engineering is a pipe filled with randomly positioned spherical obstacles, discretised with 5,210,124 fluid nodes. The square channel containing no solid fraction is made up of 5,248,000 fluid nodes.

A limitation of the preprocessor concerning the maximum size of the bounding-box made it impossible to provide larger domains, so the average number of fluid nodes per partition for the 128 CPU case was below 50,000 fluid nodes.

As can be seen from the performance results (see Fig. 3.13), the relatively small domain size for the NEC SX-8 caused a performance degeneration beyond 64 CPUs, since the number of fluid nodes per partition becomes too small for an efficient execution.¹⁵ Up to 64 CPU the speedup appears to be almost linear for all three samples. The peak performance for 128 CPUs was measured to be beyond 3000 MLUP/s for all three cases (3705 MLUP/s for the largest case, the aorta).

On the INTEL Xeon cluster, the above mentioned vector architecture typical performance loss for small sample sizes is not present, and an almost linear speedup can be seen up to 128 CPUs (see Fig. 3.14).

¹⁴The performance measurements were done by L.Axner (UvA) in close collaboration with Peter Lammers (HLRS), Thomas Zeiser (RRZE) and the author using the NEC SX-8 at HLRS Stuttgart and the INTEL Xeon cluster 'Lisa' at UvA.

¹⁵Very recent investigations [78] indicate an optimal performance on the NEC SX-8 from above several 10⁵ fluid nodes per CPU.



Figure 3.13: Performance of an MPI-parallelised Lattice-Boltzmann algorithm for three different geometries on the NEC SX-8 (performance measurements by L.Axner).



Figure 3.14: Performance of an MPI-parallelised Lattice-Boltzmann algorithm for three different geometries on the INTEL Xeon cluster (performance measurements by L.Axner).

These performance figures proof the possibility of a very good parallelisation for a sparse Lattice-Boltzmann code. For very large domains making optimal use of a 128 CPU NEC SX-8 machine, pre-processing and partitioning are the most severe challenge, since several 100 million fluid nodes have to be considered.

3.3 Effort-Efficiency Considerations

As described in the previous sections, it is possible to implement the Lattice-Boltzmann method on state-of-the-art vector and vector-parallel high performance computers with a very good efficiency. So the question arises, if this approach should always be considered when developing a Lattice-Boltzmann code.

To answer this question, several considerations play a role:

- For what purpose is the software designed: proof-of-concept, experimental, research or production?
- How much time and skilled manpower can be invested to develop the code ?
- How easy should it be to modify the code (for professionals, software experts, students, ...)?
- Which platform is available?

A vast majority of papers published in the Lattice-Boltzmann community are, if they contain simulation results at all, produced with relatively simple and straight forward implementations on a single-CPU scalar computer.¹⁶ Limitations with regard to memory or compute power play no role, since the purpose of these publications and the underlying research is to prove the applicability of a certain method, often with a simplified 2D model. If the focus is on developing a theoretical method and to give a proof-of-concept, indeed it makes no sense to invest too much time in the implementation. An easy to understand and easy to modify code serves best for such a purpose.¹⁷

Alas, more often than not, these very simple codes might sooner or later be used to do 'the real thing': larger three dimensional simulations to answer certain research questions.¹⁸ Quickly it turns out that the single-CPU desktop machine is either running out of memory, the required CPU-time is beyond acceptable limits, or both.

In the case of academic research, usually a compute-center is available, which offers access to a selection of high-performance computers virtually for free. In a worst case scenario, what was meant to be a pure test-code is now put on an expensive high-performance machine, without carrying out any algorithmic improvements. And it could happen that the CPU-power and memory of the several 100,000 Euro machine is indeed sufficient to solve the research question addressed. Although this might be a fully satisfying solution for the individual scientist interested in dealing with his research, and not with details of a performance optimised implementation, such an approach could be considered a misuse of typically public financed resources.

¹⁶Although a certain trend towards more and more technically advanced and performance optimised implementations could be observed during the past years.

¹⁷Before a selection of textbooks on Lattice-Boltzmann was available, the author implemented and put online a simple straight forward FORTRAN teaching code called 'anb' [87]. This was meant to help beginners to understand the simplest way of transforming the equations into software and the underlying algorithmic concept.

¹⁸The author got much feed-back from students 'abusing' the 'anb'-software for such purpose - against an explicit warning in the header of the code.

At this point, efficient implementation aspects come into play. Before doing the big step of re-implementing the whole code as a vector- or vector-parallel algorithm (which is indeed a major effort), more basic algorithmic and implementation improvements can be done. Simplifying algebraic expressions, taking into account loop orders and a cache-coherent memory access are easy to do but powerful measures to improve the performance even for the desktop computer.

If even these improvements can not sufficiently speed-up the code and reduce memory consumption, the implementation of a real high-performance code must be considered. Usually, it takes less time to create a well designed HPC Lattice-Boltzmann implementation with an optimised data-structure from scratch, than repeatedly manipulating the original sources.

Once this decision is taken, the design-phase is a very important step, since wrong directions here have a long-term after-effect for all researchers later involved in the development of the code. As can be seen in the previous chapters, implementing a well performing vector-parallel Lattice-Boltzmann flow-solver is feasible. If sufficient skilled and motivated manpower is available, it makes sense to consider a multi-platform implementation with hardware specific extensions which can be switched on- and off on demand.

On the other hand, it is necessary to consider that extensions to the code should still be possible, even when carried out by not so experienced students. A too highly performance optimised code often appears as a scary maze to the beginner, which does not much encourage (or simply takes too much time) to add certain extensions. A two-step solution is here usually best: provide and easy-to use interface, and once the model is working, the performance optimisation can be done by the experts.¹⁹

Concluding these remarks, a simple three-step rule for developing and porting a Lattice-Boltzmann code can be given:

- 1. Optimise for the desktop-PC.
- 2. Port- and optimise for the HPC machine.
- 3. Re-implement a performance optimised (vector-)parallel code.

3.3.1 Choosing the Hardware and Programming Language

Which platform is the best for Lattice-Boltzmann? From the author's experience,²⁰ it is a vector-computer. As can be seen by comparing the performance of the NEC SX-8 vector computer and the INTEL Xeon cluster (Fig. 3.13 and Fig. 3.14), 128 CPUs of the PC-cluster provide less MLUP/s than 8 CPUs of the vector computer. Taking into account the shared-memory access within one node (8 CPUs) of an NEC SX-8, reaching the performance of a modern medium sized PC-cluster with a vector-computer does not even require the MPI-parallelisation of the code.

¹⁹It seems to be a certain trend that research groups hire professional software engineers to help the PhD students doing the basic design and final optimisation of there codes, so they have more time to focus on their research.

²⁰On vector machines as the Convex C210, Cray Y-MP, Fujitsu VPP and NEC SX-4-8.

What if the 'ideal machine' is not available? When the problem is not easily solved by filling in the user account request to get access to a vector computer, the question of the 'best computer' is not so easy to answer. In case the hardware has to be purchased, the usually better cost/performance ratio of PC-clusters when compared with vector computers, at least for medium-sized problems where the number of CPUs can be kept within reasonable limits, plays an important role.²¹

What is saved in initial investment and gained in peak performance often has to be paid off by the researcher in additional hardware-specific programming work and a higher effort for system administration. The author experienced more than one case where an institutes small PC-cluster (if up and running at all) was not used for MPI-software, but as a rack of single-CPU machines.

Considering the power consumption is also of increasing importance. Although figures in Watt per MLUP/s are not yet available, it might be worth doing this analysis, at least when large sustained simulations are to be expected on the new machine.²²

Another question is that of the best programming language. Let aside the historical development which is responsible for the still strong bias towards FORTRAN at least for the HPC research codes, modern object-oriented languages have definite advantages for a distributed team of developers working at certain modules of the code.

It is always a challenge to find a proper balance between the easiness of implementing and maintaining a code, and the resulting performance. When an easy to maintain highly modular C++ code requires a medium-sized PC-cluster to produce a performance output equivalent to a single-CPU FORTRAN code,²³ this balance is possibly not given any more. On the other hand side a high performant and complex FORTRAN 90 code makes it possibly very difficult to carry out any extensions or modifications without understanding and modifying the whole kernel.

The discussions in the performance-aware circles of the community reveal a trend that in the future possibly a hybrid implementation would be the best: object oriented for datastructures, I/O and user interfaces, FORTRAN or C for the performance critical routines.

3.3.2 Latest Trends

A very recent and exciting development challenging the traditional domain of HPC is the promising attempt of porting Lattice-Boltzmann kernels on a graphics processing unit (GPU) [89, 90, 91, 92].²⁴ The required graphics hardware is relatively cheap, but appears

²¹This calculation does not take into account the much higher maintenance costs of a PC cluster, which has notorious hardware failures, and also not the waiting time of the scientist, until the busy system administrator has time to fix it. A vector computer's usual state, to the author's best experience, is 'up and running'.

²²The power consumption of a PC-cluster including cooling within the first 3 years is nowadays roughly the equivalent of its procurement cost [88].

²³The author has encountered more than one example of this kind.

²⁴There have been other attempts to port Lattice-Gas or Lattice-Boltzmann codes on specialised hardware or develop specific hardware for this purpose: Norman Margolus' CAM-6 [93] and CAM-8 [94], the approach of Exa Corporation to develop a specialised co-processor for the SGI workstation "which directly implements the particle collision logic" [95] or to use the VIS instruction set of the Sun Ultra SPARK CPU for this purpose [96]. All these are history, since usually the next generation general purpose CPU allowed the

to be highly suitable for Lattice-Boltzmann. An implementation by the group of IRMB (Braunschweig, Germany) [97] for a 2D simulation using the D2Q19 BGK model on a single nvidia GeForce 8800 Ultra card resulted in the impressive performance of 650 MLUP/s, and 1300 MLUP/s were achieved on a coupled system with two cards applying pthread parallelisation. For 3D simulations the D3Q13 MRT scheme attained a performance of 470 MLUP/s.²⁵

3.4 Visualisation²⁶

Using the Lattice-Boltzmann method, detailed simulations of time and space dependent, highly complex processes can be performed for hundreds of thousands of time-steps in a computational domain extending over several ten to hundred million grid points. Writing to disc the whole data set for each time-step of such a transient simulation is beyond the capacity of any storage system. To reduce the amount of data to be stored on the hard disc, significant results to answer the research question must be extracted during the simulation.

One might argue that occasional visualisation of transient flow or complex multi-physics processes is helpful to improve an understanding. Therefore an efficient approach is required for visualising the enormous amount of data produced each time step. Problems occur when visualisation of a large-scale simulation is attempted with conventional post-processing tools. First, an enormous amount of numerical data has to be transferred over the network. Moreover, huge disk space is necessary to store the data both on the computing server and the user's terminal and a large amount of memory is required to manipulate the data. One solution to these problems is server-side visualisation, where the entire image rendering process is conducted (in parallel, if required), utilising the computing server's resources. Thus, the storage and transfer of data can be significantly reduced, since only the (small) image data have to be transferred over the network.

If the visualisation chain is adequately efficient,²⁷ the opportunity of interacting with the running simulation presents. Belleman [103] defines interactive simulation environments as "dynamic systems that combine simulation, data presentation and interaction capabilities that together allow users to explore the results of computer simulation processes and influence the course of these simulations at run-time." When setting up this kind of interactive environment, the goals are "to shorten experimental cycles, decrease the cost of system

production of results with a higher performance than these specific solutions.

²⁵This is *not yet* a real competitor for traditional high-performance computing, because the memory of these cards is *currently* below 1 GB and there are certain restrictions concerning possible stencils, the domain size and other algorithmic restrictions. Once these problems are overcome and advanced Lattice-Boltzmann models with local mesh refinement and improved wall boundary conditions can be implemented on these cards, it might well be possible to build a CFD simulator for engineering applications below 10,000 Euro. Another possible application of Lattice-Boltzmann on GPUs in the area of real time processing for computer games and other related fields, see the beginning remarks of Chap. 6.

²⁶The research work presented in this section was performed at the C&C Research Laboratories, NEC Europe Ltd., (St.Augustin, Germany) in cooperation with engineers of the HPC Marketing Promotion Division, NEC Corporation, Japan. It was presented at the First International Symposium on Advanced Fluid Information - AFI-2001 (Sendai, Japan, 2001) and the Parallel Computational Fluid Dynamics - ParCFD 2002 (Kansai Science City, Japan, 2002), published in [98, 99, 100, 101].

²⁷Efficiency aspects of a real-time visualisation are addressed in [102].

resources and enhance the researcher's abilities for the exploration of data sets or problem spaces" [102].

One successful example of coupling a Lattice-Boltzmann flow solver with a visualisation and steering environment is the interactive air flow simulation within a virtual reality environment, developed at the 'Lehrstuhl für Bauinformatik' in Munich [104] and the IRMB in Braunschweig [105]. In the area of medical physics, Belleman describes an approach for the interactive simulated vascular reconstruction in a virtual operating theater [102].

3.4.1 Concurrent Visualisation System RVSLIB

The visualisation system RVSLIB,²⁸ designed for the NEC SX-series vector computers, was used within a variety of studies presented in this thesis. Although lacking advanced features such as stereoscopic visualisation in a cave, its particular performance-optimised implementation of server-sided image rendering capabilities made it a good candidate for generating movies without significant loss of performance or implementation overhead.

System Configuration

RVSLIB is a server-client type system: RVSLIB/Server, the server module of RVSLIB, runs on the computing server and is invoked by subroutine calls linked to the Lattice-Boltzmann flow solver. The transient simulation results stored in the computer's memory are directly referenced, so it is possible to run visualisation processes at a high speed and reduce the amount of necessary memory. Since the rendered image can be stored, the output of the computational results to a file can be avoided within the process of movie generation. RVSLIB/Server performs various kinds of visualisation processes and generates visualised images. These images are assembled to generate animations.

RVSLIB/Client, the client module of RVSLIB, runs on the user's desktop machine and displays a GUI for operating the system. It runs on PCs as a JAVA application or a JAVA applet.

In the interactive mode, the server compresses visualised images and sends them to the client, where they are restored and displayed on the GUI with the client module. Input visualisation parameters are sent from the client to the server and used in the visualisation processes at the next time step. Because some of the parameters are delivered to the simulation program through the server module, they can be used for controlling the simulation itself. The user can arbitrarily set such parameters by manipulating the corresponding GUI menu that will automatically appear at run time. The communication between the server and the client can be established or disconnected at any time even during the simulation. This function is particularly effective for large-scale simulations that generally take a long time. In addition to the specific communication protocol, based on sockets, the HTTP protocol can be chosen for sending data particularly through firewalls.

In the batch-processing environment, the Lattice-Boltzmann code and RVSLIB/Server run

²⁸RVSLIB is a registered trademark of NEC Corporation, Japan. The acronym is standing for Real-Time Visual Simulaton Library.

on the computing server without any communication with the terminal. In this case, when a process of changing parameter values is previously written to file as a scenario, RVSLIB automatically performs visualisation according to the scenario and outputs a series of animation data on the computing server. The scenario can easily be written in a key-frame format. If more than one scenario is prepared, animations are created independently for each scenario.

RVSLIB/Server is provided as a library format. RVS_INIT and RVS_TERM are for initialisation and termination, respectively. RVS_BFC obtains the memory addresses of the arrays are used in the simulation for storing the grid data and computational results. Adjustable arrays are supported to mitigate limitations to available data formats. RVS_MAIN communicates with RVSLIB/Client and performs various visualisation processes. RVS_BFC is for a simulation using a BFC (boundary fitted coordinates) grid.



Figure 3.15: RVSLIB GUI and a snapshot from a time dependent visualisation of the chemical reaction $A + B \rightleftharpoons C$ (volume rendered image of the product concentration [*C*]) simulated with the Lattice-Boltzmann code (for details of this simulation see Chap. 5.1).

Process Distribution

Here it is briefly described how a series of processes can be distributed between a computing server and a user's terminal for the concurrent visualisation of numerical simulation results. The main processes are the simulation, the mapping process, the rendering process, and the user interface. The mapping process generates 3D graphical objects such as polygons and poly-lines from the simulation results defined at grid points. The rendering process generates the 2D image data of the 3D graphical objects. When the simulation runs on the computing server and the user interface runs on the user's terminal, there are various possible approaches to the distribution of mapping and rendering processes [106]. RVSLIB adopts a server-side visualisation approach, in which both mapping and rendering processes are conducted on the computing server. This approach is highly tailored to large-scale simulations due to the reduced amount of data transferred over the network, thus realising practical visualisation even with low network bandwidth. If the number of grid points is $\mathcal{O}(n)$ in one spatial direction, the amount of visualised data is independent of n in the server-side visualisation, whereas it is from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$ in other approaches. This characteristic of the server-side visualisation is favourable for large-scale simulations. Moreover, with this approach a user's terminal with rather low-performance specifications can be used as a visualisation terminal. The server-side visualisation consumes CPU time on the compute server which has to be considered when running large scale simulations with concurrent visualisation.

Coupling with the Lattice-Boltzmann Code

RVSLIB visualisation routines can be integrated into a Lattice-Boltzmann code by adding only a few subroutine calls.

A snapshot of concurrent visualisation of a time dependent catalytic reaction-diffusion process in a porous media is shown in Fig. 3.15 (for details of this simulation see Chap. 5.1).

The additional CPU-time needed for image generation and data compression during the simulation is, depending on what has to be visualised, only a few percent of the total CPU-time. With the Lattice-Boltzmann algorithm, all local quantities (velocity, pressure, and concentration of chemical species) are calculated from the density distributions during the simulation procedure once per iteration. For visualisation, additional arrays need to be defined to store, and later provide, these quantities for the RVSLIB routine calls at the end of the main loop.

The memory for holding these variables²⁹ scales linearly with the size of the bounding box (not the fluid nodes) and the number of variables to be displayed. A minimum of three times the size of the bounding box for storing the mesh in terms of three Cartesian coordinates as INTEGER is required. Each flow variable (usually stored as REAL type) needs an additional array of the size of the bounding box.

This additional memory-overhead has to be considered in addition to the required extra CPU-time. In spite of these extra costs, generating movies from large scale transient simulations by this post-processing approach is feasible within a reasonable effort.

²⁹It is worth reminding that within the Lattice-Boltzmann algorithm the macroscopic primitive flow variables, the pressure and the three components of the velocity vector, are not permanently stored. They are typically computed only during the relaxation subroutine from the density distributions and stored in local variables, which are no longer needed after leaving the relaxation subroutine.

Chapter 4

Complex-Flow Studies

In this chapter, a variety of complex-flow studies are presented together with detailed validation, by comparing the results to analytical or other simulation results, and in the last section, to experimental results. The aim of this chapter is to highlight some typical applications where the Lattice–Boltzmann method is not only able to produce accurate results, but possibly advantageous compared with other CFD methods.

4.1 Channel Flow in Increasingly Complex Geometries¹

This chapter is based on an early journal publication of the author and contains one of the first quantitative comparisons between Lattice-Boltzmann and Navier-Stokes methods. A modified version of the geometry of this initial benchmark was later used as a test-case in the framework of the Deutsche Forschungsgemeinschaft (DFG) 'Lattice-Boltzmann Arbeitsgruppe', comparing the efficiency of a high-end Lattice-Boltzmann implementation with a Navier-Stokes flow solver [107].²

As a first simple test case for the comparison of the Lattice-Boltzmann method with the Finite-Volume approach we consider two dimensional flow in a channel of height H and length L = 4H, with different numbers of square obstacles regularly placed in the second quarter of the channel (see Figs. 4.1 and 4.2). The sizes of the obstacles are chosen such that the occupied space is the same for all cases and the distance between the wall and the obstacles nearest to the wall is always half the distance between the obstacles. Thus, when increasing the number of obstacles, this test case represents a flow problem of systematically increasing geometrical complexity. The increasing complexity of the flow pattern is illustrated in Fig. 4.3 which shows the profiles of the velocity magnitude in a cross section closely behind the obstacle area for the different obstacle numbers.

¹The research work presented in this section was accomplished at the LSTM Erlangen (Germany) under supervision of M.Schäfer and F.Durst. It was published in [39]. Financial support by the Bayerische Forschungsstiftung in the Bayarian Consortium of High–Performance Scientific Computing (FORTWIHR II) is gratefully acknowledged.

²Further comparison between Lattice-Boltzmann and Finite-Volume methods can be found in [108].



Figure 4.1: Channel dimensions and locations p1 and p2 for evaluating the pressure difference.



Figure 4.2: Obstacle structure with increasing complexity.

Concerning the boundary conditions at the inlet and outlet of the channel, for the Finite-Volume code a parabolic inflow profile corresponding to a Reynolds number of Re = 0.1 (based on the channel height) and a zero gradient outflow condition is chosen, whereas the Lattice-Boltzmann boundary conditions are periodic in the flow direction. A change in the density distribution of the first lattice row at the inlet (decrease in cells 2, 3 and 4, increase in cells 1, 5 and 6) leads to a directed flow for the Lattice–Boltzmann case.³ The flow

³These very simple boundary conditions are a direct extension of the re-distribution of particles in the framework of Lattice-Gas simulations (see Fig. 2.6 in Chap. 2.2.1). Much more sophistic boundary conditions are meanwhile available (see e.g. [32, 33, 34, 46, 47]). Nevertheless, the described implementation has no



Figure 4.3: Profile of velocity magnitude in a cross section closely behind the obstacle area for different obstacle numbers.

profile appeared to be parabolic at just a few lattice rows away from this first 'acceleration' row. The region downstream of the obstacle area, for the considered Reynolds number, is long enough to avoid any influence of the outlet to the flow profile.

The convergence criterion for the iteration process in the finite volume method is that the absolute sum of the residuals for mass and momentum (weighted with the corresponding inlet values) is less than 10^{-4} . For the Lattice-Boltzmann method the program stops when the maximum deviations of the mean flow rate in the last quarter section of the channel differs by less than 10^{-4} over the last 50 iterations.

As reference quantities for evaluating the accuracy of the computations, the pressure difference between the cross-sections at x = L/8 and x = 7L/8 of the channel (see Fig. 4.1) and the maximum absolute value of the velocity were computed.

In Fig. 4.4 the reference quantities are indicated for the 2 * 2 obstacle case for both methods, using lattice/grid sizes of increasing refinement ranging from 64 * 16 cells (grid 1) to 1024 * 256 cells (grid 5). For the Finite-Volume method an equidistant Cartesian grid is employed.

The results show very good agreement in the high resolution limit for both pressure drop and maximum velocity. In fact, the pressure drop on the coarser grids is closer to the fine grid value for the Lattice-Boltzmann, whereas the Finite-Volume method gives closer values for the velocity on the coarsest lattice. For the other obstacle configurations very similar results concerning the dependence of the accuracy of the reference quantities on the grid fineness were found.

In Fig. 4.5 the error in the reference quantities for a fixed grid size (256 * 64 cells) is plotted against the number of obstacles for both methods indicating the dependence of the accuracy of the methods on the complexity of the geometry. It can be observed that, concerning the

influence on the accuracy of the results, although it is less then optimal with regards to efficiency.



Figure 4.4: Pressure drop (left) and maximum velocity (right) versus grid size (indicated as lattice/grid) using Lattice-Boltzmann (LB) and Finite-Volume (FVM) methods.

pressure, the Lattice-Boltzmann results are slightly more accurate, whereas for the velocity error the Finite-Volume program is closer to the final values.⁴



Figure 4.5: Error in pressure drop (left) and maximum velocity (right) versus number of obstacles using Lattice-Boltzmann (LB) and Finite-Volume (FVM).

This interesting observation was also made (for a different flow case) by Kandhai et al. [108]. The most likely cause for this behaviour is the 'natural' derivation of the pressure from an equation of state in the case of Lattice-Boltzmann, as opposed to applying a pressure correction scheme as it is usually done within Navier-Stokes solvers. Another possible contribution comes from the higher symmetry of the Lattice-Boltzmann stencil, providing exchange also with the diagonal nodes.

⁴The big gap between the Navier-Stokes and the Lattice-Boltzmann error for the last set (right hand side of the lower Fig. 4.5), referring to the largest number of 256 obstacles in the channel, is possibly caused by a finite-size effect due to an insufficient resolution of the geometry (see Chap. 2.2.3). In the given case the number of obstacles was increased up to 16×16 , while the lattice size of $lx \times ly = 256 \times 64$ was kept constant.

4.1.1 Conclusion

This initial test case demonstrates that it is possible to produce converging results for low-Reynolds complex geometry flow with both, the Lattice-Boltzmann and Finite-Volume method. Due to its nature, the Lattice-Boltzmann method shows a slightly better mesh convergence concerning the pressure, while the Finite-Volume method has a slight advantage in computing the velocity.

4.2 Decay of Turbulence⁵

After investigating complex geometry steady-flow case in the previous example, a complex transient-flow study in simple geometries is presented in the following section.

The decay of an initial turbulent shear layer has previously been investigated by Martinez [111] and the decay of an isotropic turbulent flow field and a Taylor-Green vortex by Chen [112]. The present validation test cases were chosen in order to assess the influence of the numerical dissipation compared with the physical viscosity. This can be examined by comparing the time evolution of an initial flow field to analytical or empirical data where available.

As a simple test case for shear driven dissipation in viscous flow, the temporal evolution of a vortex described by Eqns. 4.1 and 4.2 was simulated and compared with analytical solutions. The decay of a synthetic turbulent velocity field was chosen in order to demonstrate the capability of the method for more realistic viscous fluid flow phenomena, such as turbulence.

4.2.1 Time Evolution of a Vortex

The velocity components u_0 and v_0 of an initial vortex described by the potential ψ (Eqn. 4.1) are given by Eqn. 4.2.

$$\psi(x, y, t = 0) = \psi_0 e^{-\frac{((x-x_0)^2 + (y-y_0)^2)}{a^2}}$$
(4.1)

$$u_0 = -\frac{2(y - y_0)}{a^2} \psi, \qquad v_0 = \frac{2(x - x_0)}{a^2} \psi$$
(4.2)

For this velocity field, an analytical solution (Eqns. 4.4 and 4.5) can be found in terms of acoustic Reynolds number Re_{ac} , defined in Eqn. 4.3 with speed of sound c_s , characteristic length *l* and kinematic viscosity *v* (for details of the derivation see [113]):

⁵The research work presented in this section was performed at LSTM Erlangen (Germany) under supervision of G.Brenner and F.Durst. It was presented at the 7th International Conference on the Discrete Simulation of Fluids (Oxford, UK, 1998) and published in [109]. Financial support by the Bayerische Forschungsstiftung in the Bavarian Consortium of High-Performance Scientific Computing (FORTWIHR II) is gratefully acknowledged. A recent publication by Yu et al. [110] repeated the set of test-cases in a more detailed analysis.

$$Re_{ac} = \frac{c_s l}{v} \tag{4.3}$$

$$u(x,y,t) = u_0 - \frac{2\psi_0}{a^2} \frac{1}{\alpha} \exp\left[-\gamma_x - \gamma_y + \frac{\beta_x^2}{\alpha} + \frac{\beta_y^2}{\alpha}\right] \left(y - 2\frac{\sqrt{t}}{\sqrt{Re_{ac}}}\frac{\beta_y}{\alpha}\right)$$
(4.4)

$$v(x, y, t) = v_0 - \frac{2\psi_0}{a^2} \frac{1}{\alpha} \exp\left[-\gamma_x - \gamma_y + \frac{\beta_x^2}{\alpha} + \frac{\beta_y^2}{\alpha}\right] \left(x - 2\frac{\sqrt{t}}{\sqrt{Re_{ac}}}\frac{\beta_x}{\alpha}\right)$$
(4.5)

with definition of the quantities α , β_x , β_y , γ_x and γ_y as:

$$\alpha = 1 + \frac{4t}{Re_{ac}a^2} \tag{4.6}$$

$$\beta_x = \frac{2\sqrt{t}(x-x_0)}{\sqrt{Re_{ac}}a^2} \tag{4.7}$$

$$\beta_y = \frac{2\sqrt{t}(y - y_0)}{\sqrt{Re_{ac}}a^2} \tag{4.8}$$

$$\gamma_x = \frac{(x - x_0)^2}{a^2} \tag{4.9}$$

$$\gamma_y = \frac{(y - y_0)^2}{a^2} \tag{4.10}$$

The computations were performed on a 100 * 100 lattice with periodic boundaries, which was initialised with equilibrium density distribution f_i^{eq} computed for equilibrium pressure and velocities given by Eqns. 4.1 and 4.2 for an acoustic Reynolds number $Re_{ac} = 1000$. This initial field (see Fig. 4.6)⁶ was iterated a few times to get the correct non-equilibrium part of the distribution.⁷

The square velocity and vorticity of the computed flow field were measured along the horizontal centreline of the flow field at (x = 1...100, y = 50) at t = 0,5000,10000 and 20000 iterations and compared with the analytical solutions (Eqns. 4.4 and 4.5).

As can be seen from Fig. 4.7, the computed values fit the theoretical values for the measured times. Only for the first few 100 iterations was a slight deviation from theory observed, which is obviously due to the initialisation of the densities with equilibrium distribution f_i^{eq} . This initial disturbance disappears after a short time and has no influence on the results obtained at higher iteration numbers.

⁶Since periodic boundary conditions were used, the flow field is disturbed at the domain boundaries, particularly visible in the corners. A test with a smaller lattice indicated that this effect can be neglected for the results shown in Fig. 4.7.

⁷This is a possible but not the best way to initiate a flow field, see also Chap. 6.5.1 'If I did it today ...'.



Figure 4.6: Square velocity (grey) and velocity vectors of a vortex (centre section shown here).



Figure 4.7: Time evolution of vorticity and square velocity of a vortex. Numerical results are marked by symbols, analytical results as dashed lines.

4.2.2 Decaying Turbulence Field

As a further test case, results for a temporally decaying turbulent velocity field will be presented. For that, the analysis of the properties of homogeneous and isotropic turbulence provides an analytical reference solution.

In order to provide initial conditions for the computations of a turbulent flow, an approach outlined in [113] and [114] is used, prescribing a random velocity field with the following properties: for a scalar energy spectrum according to von Karman and Pao [115], the velocity components are obtained in Fourier space assuming a homogeneous, isotropic and divergence-free vector field and a random phase angle. A fast Fourier transform is used to pass to the physical space. The parameters entering into the procedure are the Reynolds number, the initial turbulent kinetic energy k_0 and the dissipation rate ε_0 .

The computations were performed on a lattice with 513 * 513 lattice nodes with an acoustic Reynolds number of the initial flow field of $Re_{ac} = 2000$. Similar to the previous case of a vortex, the flow field was initialised with equilibrium density distribution f_i^{eq} and iterated a few times to get the correct non-equilibrium part of the distribution function.

The vorticity of the initial flow field after 0.5, 2 and 4 eddy turn-over times can be seen in Fig. 4.8. One can observe the fast decay of all small scales and the evolution of large clustered structures which fill the computational domain.



Figure 4.8: Time evolution of the vorticity of a homogeneous isotropic turbulent field for 0, 0.5, 2 and 4 eddy turn-over times.

It is known from the theory of turbulent flow that the dissipation ε and the turbulent kinetic energy *k* suffer a decay which can be described by

$$\frac{k}{k_0} = \left[1 + \frac{\varepsilon_0}{k_0}(C - 1)t\right]^{\frac{1}{1 - C}}$$
(4.11)

$$\frac{\varepsilon}{\varepsilon_0} = \left[1 + \frac{\varepsilon_0}{k_0}(C-1)t\right]^{\frac{C}{1-C}}$$
(4.12)

with a semi-empirical constant C = 2.5 for two-dimensional flows [113]. These averaged quantities were measured during computation and compared with the theoretical values. As can be seen from Fig. 4.9, the values obtained from the Lattice-Boltzmann simulation fit the theoretical curves very well.



Figure 4.9: Time evolution of dissipation ε and turbulent kinetic energy k for a homogeneous isotropic turbulent flow field. Numerical results are marked by symbols, analytical results as lines.

4.2.3 Conclusion

The results for the above test cases clearly show the possibility of performing accurate numerical simulations for viscous incompressible transient flows with the Lattice-Boltzmann method. Especially, as is known from Lattice-Boltzmann theory, no problems with numerical dissipation exist, which allows for inclusion of this quantity in a suitable definition of viscosity.

4.3 Plain Channel Flow around a Square Cylinder⁸

As a next step, in this section the combination of both, geometry in the flow field and transient flow, have been investigated by applying the Lattice-Boltzmann method to a classical benchmark case. This very detailed study clearly shows some limitations of an implementation using simple bounce-back wall boundary conditions without local mesh refinement.⁹ Nevertheless, the very good agreement with the majority of simulation results produced by a Finite-Volume code demonstrate the high accuracy of even simple Lattice-Boltzmann implementations.

The confined flow around a cylinder with square cross-section mounted inside a plane channel (blockage ratio B = 1/8) was investigated in detail by two entirely different numerical techniques, namely a Lattice-Boltzmann implementation and a Finite-Volume method. In order to restrict the approach to two-dimensional computations, the largest Reynolds number chosen was Re = 300 based on the maximum inflow velocity and the chord length of the square cylinder. The Lattice-Boltzmann code was built upon a lattice-BGK D2Q9 model. The Finite-Volume code was based on an incompressible Navier-Stokes solver for arbitrary, non-orthogonal, body-fitted grids. Both numerical methods are of second-order accuracy in space and time. Accurate computations were executed on grids with different resolutions. The results of both methods were evaluated and compared in detail. Velocity profiles and integral parameters such as drag coefficient, recirculation length and Strouhal number¹⁰ were investigated.

4.3.1 The Flow Problem

The flow past bluff bodies, especially cylinders, has been an attraction in all kinds of fluid mechanical investigations for a long time. Most of these studies were concerned with the circular cylinder case under free flow conditions. Excellent reviews on this topic were written by Williamson [118] and Zdravkovich [119]. In contrast to the overwhelming number of publications on the flow past circular cylinders, the square counterpart has not been investigated to the same extent, although it plays a dominant role in many technical applications such as building aerodynamics; for details, see, e.g., [120, 121, 122, 123, 124, 125, 126, 127]. Owing to fixed separation points for sharp-edged bodies, it is generally accepted that aerodynamic coefficients are less dependent on the Reynolds number than for circular structures.

Depending on the Reynolds number, different flow regimes can be distinguished for a square cylinder [128]. At very small Reynolds numbers (Re < 1), viscous forces dominate the flow. For this 'creeping flow', no separation takes place at the surface of the

⁸The research work presented in this section was performed together with T.Zeiser (in the framework of his 'Studienarbeit' [116] which was co-supervised by the author) at LSTM Erlangen (Germany) under supervision of M.Breuer and F.Durst. The following section is based on the publication [117]. All simulations with the LESOCC-code were accomplished by M.Breuer. Financial support by the Bayerische Forschungsstiftung in the Bayarian Consortium of High-Performance Scientific Computing (FORTWIHR II) is gratefully acknowledged.

⁹See also Chap. 6.5.1 'If I did it today ...'.

¹⁰For a definition of these quantities see Chap. 4.3.4.

cylinder. With increasing Re, the flow separates first at the trailing edges of the cylinder and a closed steady recirculation region consisting of two symmetric vortices is observed behind the body. The size of the recirculation region increases with increase in *Re*. When a critical Reynolds number Recrit is exceeded, the well known von Kármán vortex street with periodic vortex shedding from the cylinder can be detected in the wake. Different values of Re_{crit} exist in the literature. Based on experimental investigations, Okajima [124] found periodic vortex motion at $Re \approx 70$ leading to an upper limit of $Re_{crit} \leq 70$. A smaller value $(Re_{crit} = 54)$ was determined by Kelkar and Patankar [121] based on a stability analysis of the flow. When the Reynolds number is further increased, the flow separates at the leading edges of the cylinder. The onset of this phenomenon is not clearly defined in the literature; only a wide range of Re = 100 - 150 is given [124, 128]. In this Reynolds number range, the flow past square cylinders can still be considered as two-dimensional. In contrast to the circular cylinder flow for which Williamson [118] provides a Reynolds number limit of $Re \approx 180$ for the onset of three-dimensional structures in the wake, no such clear statement can be found for the square counterpart. A rough hint is given by Franke [128] with Re < 300. Therefore, this Reynolds number was chosen as the upper limit of the present two-dimensional laminar simulations. Beyond this limit three-dimensional structures have to be expected and subsequent transition to turbulence takes place in the free shear layers.

Only a few studies have dealt with the influence of confining walls on the flow phenomena around square cylinders (see, e.g., [122, 129]). In comparison with the free flow case, two new parameters have to be taken into account, the inflow profile and the blockage ratio. As shown by Davis et al. [123], the vortex shedding frequency depends strongly on the inflow profile. In the experimental investigations by Shair et al. [130] and Davis et al. [123], non negligible deviations between the velocity profiles far upstream of the cylinder and the parabolic distribution expected for fully developed laminar channel flow were observed. Therefore, this aspect has to be kept in mind for comparison between experimental and numerical investigations which typically apply the theoretical velocity profile as inflow conditions. The second parameter which plays a dominant role in confined cylinder flow is the blockage ratio of the channel, defined as B = D/H, where D is the diameter of the cylinder and H is the channel height. It is generally accepted that for a fixed Reynolds number, an increasing blockage ratio leads to an increase in the Strouhal number. This holds true for both circular and square cylinders, although the movement of the separation points cannot be responsible for this phenomenon for a sharp-edged body as assumed for a round geometry.

Davis et al. [123] investigated confined flow past square cylinders for a wide range of Re and two different blockage ratios (B = 1/6 and 1/4), experimentally and numerically. Depending on the blockage ratio, a maximum Strouhal number was observed at $Re = Re_{max} = 200 - 350^{11}$. For higher Re the Strouhal number decreases again and reaches an almost constant level. As mentioned above, non-parabolic velocity profiles were measured upstream of the cylinder. Because most numerical predictions were based on these measured inflow profiles, a direct comparison with the present study is possible only for the additional cases in which a parabolic profile was assumed. These computations were based on a Finite-Volume code and non-equidistant coarse grids of 76 * 42 and 76 * 52 grid points.

¹¹The Reynolds number $Re = Re_{max}$ is based on the maximum velocity of the inflow profile.

Two-dimensional numerical simulations were also carried out by Mukhopadhyay et al. [126] for the *Re* range 90 – 1200, two blockage ratios (B = 1/8 and 1/4) and a parabolic inflow profile. With respect to the Reynolds number of the corresponding channel flow, the upper limit of *Re* in this investigation seems to be highly questionable because turbulent flow in the channel has to be expected under these conditions. For the simulations, two different equidistant grids with 200 * 34 and 396 * 66 grid points were used. Because no clustering of grid points in the vicinity of the cylinder was applied, each surface was resolved by only 4 or 8 control volumes, respectively. As will be shown below, this resolution is definitely far too coarse to provide reliable results.

Suzuki et al. [127] performed simulations (56.3 $\leq Re \leq 225$, B = 1/20 - 1/5) on a nonequidistant grid with 207 * 54 grid points, claiming to have achieved grid independence. However, based on the present study, it is questionable that this is possible with the resolution used, especially for the higher Reynolds number cases in which separation starts at the leading edge of the cylinder. For a blockage ratio B = 1/5, Suzuki et al. [127] computed Strouhal numbers over a wide *Re* range and found a maximum at $Re = Re_{max} = 150$.

A comparison of the different data mentioned above already shows a large scatter of the results for integral parameters such as the Strouhal number (see, e.g., Fig. 4.18). There is evidently a lack of reliable experimental and numerical data for this flow case. The objective of the study presented in this section was to provide a contribution to close this gap. In order to ensure trustworthy results, two different numerical methods (Lattice-Boltzmann and Finite-Volume method) were applied and special attention was paid to the analysis of the accuracy of the solutions in terms of grid independence. For a fixed blockage ratio B = 1/8, laminar 2D flow was computed in the Reynolds number range $0.5 \le Re \le 300$. The results were evaluated in detail based on velocity fields and integral parameters and compared with previous numerical and experimental studies.

It should be stated clearly that the objective of this work was not to make a comparison of both numerical algorithms with respect to computational efficiency in terms of CPU time and memory requirements. Therefore, besides the physics of the flow past a square cylinder this section focuses on the comparison of the accuracy of both methods. Similar investigations have been reported by Eggels and Somers [131] for a non-isothermal free convective flow in a square cavity and by Eggels [132] for the direct numerical simulation of fully developed turbulent channel flow with heat transfer.

4.3.2 Finite-Volume Method

In the following sections only a brief introduction to the Finite-Volume method is given. For a more detailed description, we refer to the cited literature.

The applied code LESOCC is based on a 3D Finite-Volume method for arbitrary nonorthogonal and non-staggered grids. It was originally developed for simulating incompressible turbulent flows of practical relevance by the large eddy simulation (LES) technique [133, 134, 135, 136, 137, 138, 139, 140]. Owing to the high demands of LES with respect to spatial and temporal accuracy, the method is also well suited for the accurate computation of time-dependent laminar flows. Five different options are implemented in LESOCC for the approximation of convective fluxes. However, based on experience in
previous investigations [139], only central differences of second-order accuracy are applied for both the convective and the viscous fluxes. Time advancement is performed by a predictor-corrector scheme. A low-storage multi-stage Runge-Kutta method (three substeps, second-order accurate in time) is applied for integrating the momentum equations in the predictor step. Within the corrector step, the Poisson equation for the pressure correction (SIMPLE) is solved implicitly by an incomplete LU decomposition method. Explicit time marching works well for LES with small time steps necessary to resolve turbulence motion in time. For time accurate predictions of laminar flows, explicit time marching is the appropriate choice.

On non-staggered grids, as used in the present investigation, a special interpolation technique for the cell face velocities is necessary to prevent the decoupling of pressure and velocity components leading to non-physical oscillations. These cell face velocities are required for determining the mass fluxes at the cell faces. The momentum interpolation of Rhie and Chow [141] provides a proper coupling procedure. The influence of this approach on the solution was investigated in detail by Miller and Schmidt [142] and by Kobayashi and Pereira [143]. They found that momentum interpolation on non-staggered grids is nearly equivalent to the SIMPLE algorithm on staggered grids concerning formal error analysis and the attained accuracy of the calculations.

Of course, all models necessary to approximate the non-resolvable sub-grid scales in LES are turned off for the laminar simulations. Recently, LESOCC was extended by a multiblock structure, strongly improving the possibility of resolving complex geometries. Furthermore, the multi-block implementation was also the basis for parallelisation by domain decomposition and message passing (MPI). LESOCC is highly vectorised (vectorisation rate > 99.8%), allowing one to perform efficient computations on vector-parallel machines. Typical sustained performances are ~ 4.0 GFLOPS on four processors of a NEC SX-4 machine and ~ 3.7 GFLOPS on a Fujitsu VPP 300/700.

4.3.3 Details of the Test Case

Geometry of the Computational Domain and Grids

The 2D laminar flow around a square cylinder with diameter *D* centred inside a plane channel (height *H*) was investigated (see Fig. 4.10). The blockage ratio was fixed at B = 1/8. In order to reduce the influence of inflow and outflow boundary conditions, the length of the channel was set to L/D = 50. For the Finite-Volume computations an inflow length of l = L/4 was chosen. For the Lattice-Boltzmann simulations, the inflow length was varied between l = L/4 and L/3 to investigate the influence of different inflow and outflow lengths. However, only negligible deviations in the results were found.

The geometry for the Lattice-Boltzmann method was created with the marker-and-cell approach: single lattice nodes are either occupied by an elementary obstacle or they are free. The Finite-Volume code is written for general body-fitted curvilinear coordinates. Owing to the specific geometry in the present study, only Cartesian grids are applied. However, in contrast to Lattice-Boltzmann, Finite-Volume allows the application of non-equidistant stretched grids. This has the advantage that grid points can be clustered in regions of large



Figure 4.10: Definition of the geometry and integration domain.

gradients (e.g. in the vicinity of the cylinder) and coarser grids can be used in regions of minor interest. Consequently, fewer grid points are necessary for the Finite-Volume simulation to achieve the same accuracy as the Lattice-Boltzmann method. Table 4.1 gives an overview of all grids used in the present study. Four different equidistant grids with up to 640,000 lattice nodes were generated for the Lattice-Boltzmann simulations. The number of lattice nodes on each side of the square cylinder varies between 10 and 40, leading to smallest distances between lattice nodes of 0.1D - 0.025D. Three different grids were used for the Finite-Volume simulation. The first is equal to the coarsest grid for Lattice-Boltzmann. The second and third are stretched grids (geometrical series) where the grid points are highly clustered in the vicinity of the cylinder. On the finest grid each face of the cylinder is discretised by 100 control volumes (CV) and the smallest CV has a chord length of 0.01D, which is 2.5 times smaller than on the finest grid used for Lattice Boltzmann.

Method	Total no.	Grid	No. of CV	Smallest CV	Max. stretching
	of CV	type	on cylinder	at cylinder/D	factor
LB	500 * 80	equid.	10	0.1	1
LB	1000 * 160	equid.	20	0.05	1
LB	1500 * 240	equid.	30	0.033	1
LB	2000 * 320	equid.	40	0.025	1
FV	500 * 80	equid.	10	0.1	1
\overline{FV}	400 * 240	non-equid.	80	0.01	1.03
\overline{FV}	560 * 340	non-equid.	100	0.01	1.02

Table 4.1: Overview of all grids used.

Boundary Conditions

Wall Boundary Conditions There is a long ongoing discussion on the proper use of wall boundary conditions within the framework of Lattice-Boltzmann (see Chap. 2.2.3). It is often argued that the so-called 'bounce-back' wall boundary conditions, which are also used in the present implementation of the Lattice-Boltzmann scheme, are of first-order accuracy, whereas the Lattice-Boltzmann equation is of second-order. However, as explained in Chap. 2.2.3, more detailed investigations showed that the error produced by the bounce-back boundary condition is sufficiently small if the relaxation parameter ω is sufficiently close to 2, allowing precise knowledge of the wall position with zero flow velocity. Therefore, we believe that the bounce-back conditions can be used without any influence on the order of the Lattice-Boltzmann scheme for the square geometries considered here, if ω is chosen within a suitable range.

In the framework of Finite-Volume simulations of laminar flows, solid walls can be easily modelled by Stokes' no-slip wall boundary condition assuming $\vec{v} = 0$ at the wall. In contrast to the bounce-back condition for Lattice-Boltzmann, there is no question about the no-slip condition for incompressible flows.

Inflow Boundary Conditions In order to simulate fully developed laminar channel flow upstream of the square cylinder, a parabolic velocity profile with maximum velocity u_{max} was prescribed at the channel inlet. This velocity was chosen to be lower than 10 % of the speed of sound for the Lattice-Boltzmann simulations to avoid significant compressibility effects, which are known to increase with the square of the Mach number [144]. In the Lattice-Boltzmann implementation for this study, the pressure at the inlet was extrapolated upstream, and the equilibrium density distribution was computed from this pressure and the given velocity and imposed at the first lattice column. The inlet region was chosen to be long enough to ensure that a slight error which occurs from neglecting the non-equilibrium part in the density distribution has no influence on the results presented thereafter. The Finite-Volume code does not require any boundary condition for the pressure.

Outflow Boundary Conditions At the outflow boundary slightly different boundary conditions are used for Lattice-Boltzmann and Finite-Volume methods. However, owing to the extremely large integration domain behind the cylinder, no influence is expected for the solution in the vicinity of the cylinder. For Lattice-Boltzmann, a fixed pressure is imposed in terms of the equilibrium distribution function at the outlet. For this task, the velocity components are extrapolated downstream. For the Finite-Volume code a convective boundary condition given by

$$\frac{\partial u_i}{\partial t} + u_{conv} \frac{\partial u_i}{\partial x} = 0 \tag{4.13}$$

is used at the outflow boundary, where u_{conv} was set equal to the maximum velocity u_{max} of the inflow profile. This condition ensures that vortices can approach and pass the outflow boundary without significant disturbances or reflections into the inner domain. In all

previous computations of different test cases [133, 134, 135, 136, 139, 140], the convective boundary condition was found to work very well. Likewise, no difficulties were observed in the case of the square cylinder flow.

4.3.4 **Results and Discussion**

A Reynolds number range $0.5 \le Re \le 300$ was investigated numerically, where *Re* is based on the cylinder diameter *D* and the maximum flow velocity u_{max} of the parabolic inflow profile. The following section starts with a description of the different flow patterns observed with increasing *Re*. The subsequent sections present a detailed comparison of the computed results (Lattice-Boltzmann and Finite-Volume) based on velocity profiles at several positions in the flow field. Furthermore, the computations are analysed and compared regarding integral flow parameters such as recirculation length, Strouhal number and dimensionless force coefficients (lift and drag).

Flow Pattern

Fig. 4.11 shows computational results (Finite-Volume) in the vicinity of the cylinder by streamlines at four different Reynolds numbers (Re = 1, 30, 60, 200), each characterising a different flow regime. At low $Re \leq 1$, the creeping steady flow past the square cylinder persists without separation (Fig. 4.11(a)). The magnitude of viscous forces decreases with increasing Re until a certain value, at which separation of the laminar boundary layers occurs. In comparison with the circular counterpart, for which a value of $Re \approx 5$ was found [119], separation at the trailing edges of the sharp-edged body can be observed at lower Re. Above this limit, the wake comprises a steady recirculation region of two symmetrically placed vortices on each side, as shown in Fig. 4.11(b) at Re = 30, the length ow which grows as Re increases. The same trend was observed for circular cylinders. Owing to the sharp corners, the separation point is fixed at the trailing edge and the flow is attached at the side walls. The steady, elongated and closed near-wake becomes unstable when $Re > Re_{crit}$ (Fig. 4.11(c)). The transverse oscillation starts at the end of the nearwake and initiates a wave along the trail. This phenomenon is visualized by streak-lines in Fig. 4.12(a) (Re = 60). Weightless particles released at different sources in front of the cylinder were integrated during the time-dependent flow computation. For the circular cylinder the onset of the wake instability was found to be a manifestation of a Hopf bifurcation [118], and there is no counter-argument available to indicate that the same mechanism is not responsible for the onset of the wake instability also for a square cylinder. As stated in the introduction, Kelkar and Patankar [121] determined a critical value of $Re_{crit} = 54$. Although this limit depends on flow parameters such as the blockage ratio, a similar value $(Re_{crit} \approx 60)$ was observed in the present computations. When Re is further increased, the free shear layers begin to roll up and form eddies as shown in Fig. 4.12(b) at Re = 100. This phenomenon is well known as the von Kármán vortex street. The wavelength of vortex shedding decreases with rising Re, as seen in Figs. 4.12(a-c). Another important change in the flow structure is observed in the range Re = 100 - 150, where separation starts at the leading edge of the cylinder (Fig. 4.11(d), Re = 200). As will be seen below, this strongly influences the frequency of vortex shedding. The upper limit of this laminar 2D shedding has an enormous spread in the literature. Preliminary three-dimensional simulations with the LESOCC-code have shown that the flow computation shown in Fig. 4.12(d) at Re = 300 is slightly beyond the limit where two-dimensional simulations can be performed. The deviations from fully periodic structures in the far wake are also a clear hint for this statement. Furthermore, it should be considered that the Reynolds number based on the channel height *H* and the mean velocity u_{mean} in the channel is already $Re_{channel} = 1600$ for this case. Therefore, owing to the triggering effect of the obstacle on the channel flow, transition to turbulence has to be expected leading to three-dimensional structures in the wake.



Figure 4.11: Streamlines around the square cylinder for different Reynolds numbers, from top left to lower right: Re = 1, 30, 60, 200.

Steady Flow: $0.5 \le Re < 60$

Recirculation Length The length of the closed near-wake (L_r) has been measured for a circular cylinder, eliminating the effect of blockage by extrapolating the measured data to $B \rightarrow \infty$. Then the empirical relationship is linear [119]:

$$L_r/D = 0.05 \, Re \qquad \text{for } 4.4 < Re < 40 \tag{4.14}$$

Fig. 4.13 illustrates the computed values for the recirculation length L_r for a square cylinder inside a channel as a function of the Reynolds number. First, in Fig. 4.13(a) the Finite-Volume results obtained on the three different grids are compared in order to prove grid independence. On the coarsest grid the recirculation length is slightly shorter than that



Figure 4.12: Streak-lines around the square cylinder for different Reynolds numbers, from top to bottom: Re = 60, 100, 200, 300.

on the medium and fine grid. The agreement between the two non-equidistant grids is excellent over the entire Reynolds number range. No improvements can be expected by further grid refinement. Fig. 4.13(b) shows a comparison of the Lattice-Boltzmann and Finite-Volume results based on the finest grids used for both methods (see Table 4.1). The computed values for L_r coincide, showing a linear dependence on the Reynolds number. Similar to the relationship (Eqn. 4.14) for the circular cylinder, a curve fit of the square cylinder results (B = 1/8) leads to

$$L_r/D = -0.065 + 0.0554 \, Re \qquad \text{for } 5 < Re < 60 \tag{4.15}$$

which is also plotted in Fig. 4.13(b). As a consequence, the recirculation length of the confined square cylinder flow is slightly shorter for Reynolds numbers below $Re \approx 12$ and larger above this value in comparison with the circular counterpart. Unfortunately, no experimental data for comparison can be found in the literature for the square cylinder.

Drag Coefficient One of the most important characteristic quantities of flow around a cylinder is the drag coefficient C_d . In the region of small Reynolds numbers, the drag coefficient varies strongly with *Re*. The contributions of the viscous and pressure forces to the total drag are of the same order of magnitude. A comparison of the computed Finite-



Figure 4.13: Computed recirculation length L_r vs. Reynolds number. (left:a) Comparison of different Finite-Volume results; (right:b) comparison of Finite-Volume and Lattice-Boltzmann results on finest grids.

Volume results on the three different grids is shown in Fig. 4.14(a) for the steady-state results in the range $0.5 \le Re \le 60$. On the coarsest grid the drag coefficient is slightly smaller than on the medium and fine grid, especially at the lower end of the *Re* range. The agreement between the results of the two non-equidistant grids is excellent over the entire Reynolds number range. As mentioned previously, no improvements are expected on further grid refinement. Fig. 4.14(b) shows a comparison of the Lattice-Boltzmann and Finite-Volume data on the finest grid levels for $10 \le Re \le 60$. Deviations occur for small Reynolds numbers, but the agreement for the upper *Re* range considered is satisfactory. Because the discrepancies are larger in the lower *Re* range, where the viscous forces play a dominant role for the drag, it can be concluded that an insufficient resolution of the boundary layers by the Lattice-Boltzmann method is responsible. This also agrees with the observations based on the Finite-Volume results on the coarsest grid, which show larger deviance from the fine grid solutions at the lower end of the *Re* range.

Unsteady Flow: $60 \le Re \le 300$

Velocity Profiles In order to make a detailed comparison of the Lattice-Boltzmann and Finite-Volume results, velocity profiles at different positions in the flow field were extracted at Re = 100. Because the flow is unsteady at this Re, it was first necessary to define the moment of evaluation. In the present study this is given by the time level at which the cross-stream velocity V at an axial position of 10D behind the cylinder (x = 10.5, y = 0) changes its sign from minus to plus. Fig. 4.15 shows the distribution of the two velocity components along the centreline. Both Lattice-Boltzmann and Finite-Volume results were achieved on the finest grid (Table 4.1). The agreement between the Lattice-Boltzmann and Finite-Volume simulations is excellent in the upstream region, in the vicinity of the cylinder and also in the downstream region up to about 12D. In the far wake > 12D, small deviations occur. However, these can be explained. Owing to the stretched grids



Figure 4.14: Computed drag coefficient C_d vs. Reynolds number for steady flow. (left:a) Comparison of different Finite-Volume results; (right:b) comparison of Finite-Volume and Lattice-Boltzmann results on finest grids.

used in the Finite-Volume computations, the resolution in the vicinity of the cylinder is higher than that for the Lattice-Boltzmann simulations, whereas in the far wake the grid is much coarser. This is a typical configuration when flows around bodies are investigated and special attention is paid to the vicinity of the structure. Owing to the present version of the Lattice-Boltzmann (equidistant grids), this strategy was not applied, leading to a higher resolution and therefore more accurate results for Lattice-Boltzmann compared with Finite-Volume.



Figure 4.15: Comparison of instantaneous Lattice-Boltzmann and Finite-Volume results at a certain moment (see text for explanation): (left:a) stream-wise (U) and (right:b) cross-stream (V) velocities along the centreline (y = 0), Re = 100.

Fig. 4.16 shows velocity profiles of U and V at three different axial positions, x = 0D, 4D and 8D. For the profile through the centre of the cylinder (x = 0), no deviations are visible between the two sets of results. Further downstream small deviations occur, especially for the V component which is smaller than U and therefore more difficult to predict correctly. When assessing the agreement between the results of the different numerical methods, the unsteady nature of the flow must be considered. The definition of the moment of evaluation has a strong influence on the results. Owing to finite time steps (and also finite spatial resolution), the accuracy in time in the worst scenario is one time step size. Therefore, an exact agreement between the computational results cannot be expected.



Figure 4.16: Comparison of instantaneous Lattice-Boltzmann and Finite-Volume results at a certain moment (see text for explanation): (left:a) stream-wise (U) and (right:b) cross-stream (V) velocities at three different positions in the flow field, centre of cylinder (x = 0), near-wake (x = 4D) and far-wake (x = 8D), Re = 100.

Strouhal Number One important quantity considered in the present analysis is the Strouhal number *St*, computed from the cylinder diameter *D*, the measured frequency of the vortex shedding *f* and the maximum velocity u_{max} at the inflow plane:

$$St = \frac{fD}{u_{max}} \tag{4.16}$$

The characteristic frequency f was determined by a spectral analysis (fast Fourier transformation, FFT) of the time series of the lift coefficient C_l . Fig. 4.17 illustrates the computational results of both methods, where the Lattice-Boltzmann data are represented by lines and the Finite-Volume results are given by symbols (partly with lines). All simulations, including those on the coarse grids, agree fairly well in the *Re* range $60 \le Re \le 133$, showing an increase in the Strouhal number with increasing *Re*. At the upper limit of this range an important change in the flow structure takes place, namely the movement of the separation point from the trailing edge to the leading edge of the square cylinder. As expected, the separation on the side walls is strongly influenced by the resolution in the vicinity of the body. Therefore, the results of both methods on the coarsest grid with only 10 points on each surface do not have to be taken seriously. At the finest resolution of the Lattice-Boltzmann simulation, each side of the cylinder is represented by 40 nodes with the smallest distance to the wall being 0.025*D*. The Finite-Volume applies 100 CV at each surface on the finest level with a 2.5 times smaller wall distance (stretched grid), resulting in a much finer resolution in the vicinity of the cylinder. Therefore, the small discrepancies between the computed Strouhal numbers of Lattice-Boltzmann and Finite-Volume at Re > 133 are probably due to the insufficient resolution of the recirculation regions at the side walls for Lattice-Boltzmann. However, owing to resource limitations, no further refinement was possible for the Lattice-Boltzmann simulations. The Strouhal number has a maximum at about Re = 150 - 160 and decreases again for higher Re. With the exception of the results on the coarsest grids, the agreement between the Lattice-Boltzmann and Finite-Volume data is reasonable.



Figure 4.17: Computed Strouhal numbers vs. Reynolds number for Lattice-Boltzmann and Finite-Volume on different grids.

In order to demonstrate this good agreement in comparison with data from the literature, Fig. 4.18 provides a collection of different numerical investigations for blockage ratios B = 0 - 1/4. No experimental data are available for a parabolic inflow profile. Mukhopadhyay et al. [126] carried out numerical simulations in the *Re* range $90 \le Re \le 1200$ with equidistant grids of up to 396 * 66 CV, resulting in an extremely coarse resolution of 8 * 8 grid points on the side walls of the cylinder. For B = 1/8 the Strouhal numbers increase with increasing in *Re* without showing a maximum. In comparison with the present results included in Fig. 4.18, the Strouhal numbers of Mukhopadhyay et al. [126] are much smaller. However, the *St* values for high *Re* are particularly questionable because the flow should be turbulent inside the channel at the upper limit of the computed *Re* range. For B = 1/4 the *St* curve looks totally different and has a maximum at $Re \approx 200$. Suzuki et al. [127] also carried out numerical investigations. They applied a non-equidistant grid with 207 * 54 CV and three different blockage ratios (B = 1/20, 1/10, 1/5) were evaluated.



Figure 4.18: Comparison of computed Strouhal numbers (Lattice-Boltzmann and Finite-Volume) on finest grids with data from the literature. (left:a) Entire St(Re) range; (right:b) Zoom of the interesting region, same legend as in (a).

However, only for B = 1/5 are enough values available to determine the St(Re) relationship plotted in Fig. 4.18. The curve has a maximum at $Re \approx 150$ and is in good agreement with the results of the present study, although the blockage ratio is higher. Davis et al. [123] investigated this flow problem experimentally and numerically for B = 1/6 and 1/4. However, in the experimental investigations, non-parabolic inflow profiles were detected. For the numerical simulations, only one St value is given for each blockage ratio in the Re range of the present work. Franke et al. [120] computed the laminar cylinder flow under free stream conditions (B = 0). They found a similar St(Re) curve to that in the present investigation with a maximum at $Re \approx 150$. The St values are slightly higher than the Lattice-Boltzmann and Finite-Volume results in the present study (B = 1/8), although it is well known that an increase in the blockage ratio should lead to an increase in St^{12} . Furthermore, Okajima [124] found a local maximum of the Strouhal number at the same Reynolds number as Franke et al. [120] and the present study in his experimental investigation on rectangular cylinders under free stream conditions. In conclusion, the St data for confined square cylinder flow taken from the literature are highly scattered. The corresponding values for free stream conditions show at least a local maximum at the same Reynolds number as in the present work, but the values cannot be compared, directly owing to the influence of blockage. However, in the view of these large deviations, the differences between the computed Lattice-Boltzmann and Finite-Volume results are only marginal.

Drag and Lift Coefficients In the unsteady 2D flow regime ($60 \le Re \le 300$) the nearwake becomes unstable and a sinusoidal oscillation of the shear layers commences, later forming the von Kármán vortex street. In Fig. 4.19(a), the time-averaged drag coefficients

¹²Note that in the present study Re and St are based on the maximum flow velocity of the parabolic inflow profile and that a redefinition based on the mean velocity would change the comparison with the free stream case.

in this *Re* range are plotted for the three Finite-Volume computations. The $C_d(Re)$ curve has a local minimum at $Re \approx 150$. Up to $Re \approx 100 - 150$, good agreement is achieved on the three different grids. However, for larger Re, the C_d values on the coarsest grid deviate strongly from the data on the two finer grids, which are themselves in close agreement. This discrepancy is clearly caused by the insufficient resolution of the cylinder vicinity for the coarse grid, which plays a dominant role especially in the *Re* range at which separation moves from the trailing to the leading edge of the cylinder. Fig. 4.19(b) shows a comparison of the Lattice-Boltzmann and Finite-Volume results both obtained on the finest grid level. The agreement between the Lattice-Boltzmann and Finite-Volume data is satisfactory up to $Re \approx 100$. Above this value, the drag coefficients computed by Lattice-Boltzmann are systematically higher. Comparison of Fig. 4.19(a) and (b) shows that the Lattice-Boltzmann data on the finest grid are in close agreement with the Finite-Volume on the coarsest grid. Therefore, the deviations between Lattice-Boltzmann and Finite-Volume on the coarsest grids are expected, again, to be an effect of insufficient resolution for Lattice-Boltzmann, especially in the vicinity of the cylinder.



Figure 4.19: Computed time-averaged drag coefficient C_d vs. Reynolds number Re for unsteady flow. (left:a) Comparison of different Finite-Volume results; (right:b) comparison of Finite-Volume and Lattice-Boltzmann results on finest grids.

No experimental or other numerical data for comparison were found in the literature for the same inflow conditions and blockage ratio. However, at least the computations of Franke [120, 128] for a square cylinder under free stream conditions confirm our finding of a local C_d minimum approximately at the Reynolds number where separation is initiated at the leading edge.

Finally, in Fig. 4.20 the variation of the drag coefficient $(max(C_d) - min(C_d))$ and the lift coefficient $(max(C_l) - min(C_l))$ are plotted for the Finite-Volume computations. The amplitudes of the C_l oscillations are approximately one order of magnitude larger than the corresponding drag values. The drag variation increases progressively over the entire *Re* range, whereas for the lift variation a degressive increase is observed up to $Re \approx 150$ followed by an inflexion point. The results on the two finer grids are in close agreement. No reasonable results are obtained on the coarse grid above the inflexion point owing to the

resolution problem discussed previously. No data were found in the literature for the variation of drag and lift of confined square cylinders. Only the work of Franke [128] reports a similar curve for the amplitude of the lift coefficient, but for free stream conditions, which make a direct comparison impossible.



Figure 4.20: Variation of force coefficient vs. Reynolds number for Finite-Volume results. (left:a) Drag variation, $max(C_d) - min(C_d)$; (right:b) lift variation, $max(C_l) - min(C_l)$.

4.3.5 Conclusion

The lack of accurate and detailed data found in the literature for confined laminar flow past a square cylinder initiated the present work. In order to generate reliable numerical results, two different approaches were applied to investigate the two-dimensional flow past a square cylinder inside a channel (B = 1/8) for the Reynolds number range 0.5 < Re < 300. For both methods (a Lattice-Boltzmann implementation developed for equidistant orthogonal lattices and a general-purpose Finite-Volume method) grid independence of the results was first investigated. For steady flow (Re < 60) excellent agreement between the Lattice-Boltzmann and Finite-Volume results was found for the length of the recirculation region. Small deviations were detected for the drag coefficients in the lower Re range. The unsteady flow computations impressively demonstrate the capability of the Lattice-Boltzmann method to deal with instantaneous flows. Velocity profiles at different locations in the flow field (Re = 100) were evaluated and compared with the Finite-Volume data, showing very good agreement. Strouhal numbers were determined for the entire Reynolds number range. Both methods provide a local maximum of St at $Re \approx 150$. Compared with the scattered data in the literature, the deviations between the Lattice-Boltzmann and Finite-Volume results are almost negligible. Finally, drag coefficients were computed and compared. As is known from the literature for square cylinders in free stream, the drag coefficient of a confined cylinder also shows a local minimum at $Re \approx 150$. In conclusion, the study presented in this section provides reliable and accurate results for confined cylinder flow which were not previously available.

4.4 Numerical Analysis of the Pressure Drop in Porous Media Flow¹³

The final flow study in this chapter brings a comparison of Lattice–Boltzmann simulations to experimental results in the very domain of Lattice-Boltzmann: the simulation of low-Reynolds complex geometry flow.

The Lattice-Boltzmann method is used in this study for a detailed investigation of the origins of the pressure drop in porous media flow. In agreement with the experimental results [14] it is shown that the elongation and contraction of fluid elements is an important factor for the pressure loss in porous media flow and that a significant error is made when only shear forces are taken into account. To obtain the geometry information of the porous media for our simulations, we used the 3D computer tomography technique.

As shown in the previous chapters, one advantage of the Lattice-Boltzmann method is its specific way of handling large computational meshes regardless of the complexity of the geometry. The detailed discretisation of the porous geometry allows the exact simulation of the transport of mass and momentum without any of the underlying semi-empirical homogenisation models, generally used in engineering applications. Thus, on one hand, they allow for investigations of the transport phenomena in porous media and improvements of the basic understanding, e.g. of the high viscous pressure losses in these flows. On the other hand, valuable information entering in the formulation of homogenisation models can be obtained from these data. In this respect, the Lattice-Boltzmann method may be considered to replace standard experiments to determine quantities such as the permeability, with the additional advantage that more information about local flow properties can be obtained than usually possible in experiments.

Quantitatively accurate Lattice-Boltzmann simulations for the specific case of fibrous media have been carried out by Kandhai and Koponen et al. [45, 148]. In these studies, the permeability was measured as a function of porosity and the Lattice-Boltzmann method was shown to deliver accurate results over a wide range of porosities for sufficient discretisations, taking into account finite-size effects. It is worth noting that for high porosities the simulations showed a tendency to over-estimate the dissipative effects when compared to theoretical values. A possible explanation (in addition to the one given in the quoted literature) might well be found in the analysis presented within this section.

The standard homogenisation approaches usually applied for Finite-Volume simulations in complex geometries are based on the assumption of a linear or quadratic relationship between the pressure gradient and the mean velocity (Darcy or Forchheimer law, see e.g. [149, 150, 151, 152]). Usually, a proportionality is assumed with a constant permeability. This parameter has to be estimated from experiments or from theoretical considerations, such as

¹³The research work presented in this section was performed at LSTM Erlangen (Germany) under supervision of G.Brenner and F.Durst. It was presented at the Workshop on Scientific Computing in Chemical Engineering II (Hamburg, Germany, 1999), the International Symposium on the Discrete Simulation of Fluid Dynamics - LGA'99 (Tokyo, Japan, 1999) and the First International Conference on Computational Fluid Dynamics - ICCFD (Kyoto, Japan, 2000), published in [145, 146, 147, 57]. The author would like to thank the Hattinger Prüf- und Entwicklungs- GmbH (HAPEG) for providing the computer tomography data. This project was founded by the Deutsche Forschungsgemeinschaft (Proj.Nr. Br 1864/1).

the Kozeny-Darcy equation (see, e.g. [153]). In this theory, it is assumed that the porous media can be modelled as a bundle of capillaric tubes. Only shear forces in a laminar Poiseuille-like flow are taken into account and any forces due to elongation and contraction of fluid elements are neglected. Durst [14] demonstrated that this approach leads to an underestimation of the momentum loss by a factor of 2.5 and a significant discrepancy from experimental results for specific geometries. The discrepancy between the pressure loss predicted by such capillaric models and the experimental results is usually explained by the introduction of another fitting parameter, the 'tortuosity factor'. While in some models the tortuosity is assumed to be just a numerical parameter to fit the experimental data, other approaches link it to the effect of additional length of the channels due to the complex tortuous structure of the flow paths.¹⁴

In this last section, the results of detailed numerical simulations are used to investigate quantitatively the effect of elongational forces and their contribution to the pressure loss in porous media flows.

In the next section, we summarise the basic idea of the capillaric theory in comparison to some experimental results and present two simulations of porous media flow. In the last part, the results of these two simulations are evaluated with regard to shear and elongational forces and their contribution to the pressure drop.

4.4.1 Analytical Models for the Pressure Drop

In general, the idea behind analytically estimating a viscous porous media flow is to define a relation which describes the pressure drop as a function of the geometry (e.g. porosity, specific surface), fluid parameters (density, viscosity) and flow parameters (velocity).

$$\partial_x P = f(geometry, fluid, flow)$$
 (4.17)

Kozeny-Darcy Equation

The most common models which address this problem can be summarised under the term 'capillaric theories', where the porous media flow is being modeled as a flow through a bundle of channels with weakly changing cross-sections. For each of these channels, the Navier-Stokes equation can be solved, and the relationship of the mean flow velocity \tilde{U}_x and the pressure drop for a single channel results in:

$$\tilde{U}_x = -\frac{1}{2\mu} \left(\frac{\mathrm{d}P}{\mathrm{d}x}\right) R_h^2 \tag{4.18}$$

where *P* is the pressure, μ the fluid viscosity and *x* the mean flow coordinate. The hydraulic radius R_h , which is defined as the ratio of fluid volume and wetted surface, can be derived from the radius *R* of the pipe by:

¹⁴For a detailed discussion of the various definitions of tortuosity see e.g. [154].

$$R_h = \frac{\pi R^2}{2\pi R} = \frac{R}{2} \tag{4.19}$$

Now, assuming a bundle of tubes with an average hydraulic radius $\tilde{R_h}$ and a length L, (Eqn. 4.18) can be rewritten as:

$$\tilde{U}_x = -\frac{1}{2\mu} \frac{\Delta P}{\Delta L} \tilde{R}_h^2 \tag{4.20}$$

Neglecting the underlying channel-geometry, one can try to use this formula as a general expression for calculating the pressure drop in other types of geometries, assuming that the average hydraulic radius is known or can be derived. For example, consider a porous media built up of spheres of an average diameter $\tilde{D_p}$. $\tilde{R_h}$ can be written as a function of $\tilde{D_p}$ and its porosity ε :

$$\tilde{R_h} = \frac{\tilde{D_p}}{6} \frac{\varepsilon}{(1-\varepsilon)}$$
(4.21)

Inserting this expression into Eqn. 4.20 yields:

$$\tilde{U}_x = \frac{1}{\varepsilon} U_0 = \frac{1}{2\mu} \frac{\Delta P}{\Delta L} \frac{\tilde{D}_p^2}{36} \frac{\varepsilon^2}{(1-\varepsilon)^2}$$
(4.22)

where U_0 is the so called 'effective velocity' inside the porous media.

Eqn. 4.22 can be rewritten in so called Ergun coordinates (see, e.g. [153]),

$$72 = \left[\frac{\Delta P}{\Delta L}\frac{\tilde{D_p}}{\rho U_0^2}\frac{\varepsilon^3}{(1-\varepsilon)}\right] \left[\frac{U_0\tilde{D_p}\rho}{\mu (1-\varepsilon)}\right]$$
(4.23)

Introducing the dimensionless quantities Reynolds number Re and friction factor f as defined by

$$f = \frac{\Delta P}{\Delta L} \frac{\tilde{D_p}}{\rho U_0^2} \frac{\varepsilon^3}{(1 - \varepsilon)}$$
(4.24)

$$Re = \frac{U_0 \tilde{D_p} \rho}{\mu \left(1 - \varepsilon\right)} \tag{4.25}$$

the advantage of using the Ergun coordinates becomes obvious, because Eqn. 4.23 can be written in the compact form

$$f = \frac{\Lambda_{\rm th}}{Re} \tag{4.26}$$

with the 'theoretical' friction coefficient $\Lambda_{th} = 72$.

4.4.2 Experimental Results

The friction coefficient was experimentally measured by Durst [14] for a packed bed of spheres with different diameters with a wide range of Reynolds numbers (see Fig. 4.21).



Figure 4.21: Friction coefficient versus Reynolds number (section from [14]).

For Reynolds numbers below Re = 1, the experimentally determined friction coefficient appears to be constant with $\Lambda_{exp} = 182$. This value is about 2.5 times higher than the theoretical one derived within the capillaric theory.

To explain this additional pressure loss, it is usually argued, that the capillaric theories do not take into account the complex paths, the fluid normally has to go through a porous media. When only the effects of longer flow paths compared to the edge length of a porous media are considered, the tortuosity factor is introduced as follows (see, e.g. [153]):¹⁵

$$\tau = \frac{\text{length of flow paths}}{\text{macroscopic length scale}}$$
(4.27)

It might be doubted that a tortuosity factor of $\tau = 2-3$ only caused by the additional length of the flow paths is a realistic assumption, because this would imply that the length of the fluid channels is up to three times larger than the length of the porous media.¹⁶

In the following section, by applying the Lattice-Boltzmann method for a detailed investigation of complex geometry flow, we will show that there exists another, by the capillaric theories, not recognised physical effect causing pressure drop in porous media flow.

¹⁵As Koponen [154] remarks that "it is evident that, as a physical quantity, tortuosity is not uniquely defined".

¹⁶Direct measurements of the tortuosity for 2D samples of randomly placed rectangles were carried out by Koponen [154]. A relation between porosity and tortuosity was established, resulting in a maximum tortuosity of $\tau < 1.6$ for the lowest porosities.

4.4.3 Numerical Simulation of Porous Media Flow

The flow field produced by a Lattice-Boltzmann flow simulation through a packed bed of spheres was analysed with respect to the elongational stresses, which can produce additional pressure losses.

Boundary Conditions

A parabolic velocity inlet profile and fixed pressure at the outlet were chosen for all test cases. This was achieved by introducing the equilibrium density distribution at the first and last lattice column computed with an upstream extrapolated pressure and a downstream extrapolated flow velocity for the inlet and outlet respectively. The inlet and outlet region was chosen to be long enough to prohibit any errors introduced by this method from affecting the measured quantities.

To ensure a fix position of the solid surface with the standard bounce-back wall boundary condition applied here (see the discussion in Chap. 2.2.3), the relaxation parameter ω was only allowed to vary in a very small range after calibrating.¹⁷

Validation

To produce quantitatively reliable CFD-data, it is necessary to ensure the grid independence of the numerical results. This is usually done by discretising the same geometry with meshes of increasing sizes and observing the convergence of the results with increasing mesh refinement. When applying the marker-and-cell approach, a discretisation of spherical objects with rectangular elements makes it necessary to carefully investigate the discretisation error.

As a validation test case, the pressure drop for low Reynolds number flow through an orthorhombic package of spheres was simulated. The domain sizes were chosen to be lx * ly * lz = 120 * 20 * 20 lattice nodes for the 8 * 2 * 2 spheres with a diameter of $D_p = 10$ for the coarsest resolution and lx * ly * lz = 480 * 80 * 80 lattice nodes with $D_p = 40$ for the finest resolution. Periodic boundary conditions were applied normal to the main flow direction to make this test case similar to the experimental set up of Durst et.al. [14].

Good convergence of the numerically achieved friction coefficient to the experimental values (see Fig. 4.21) can be observed in Fig. 4.23. A sphere diameter of $D_p = 20$ is sufficient to approach the convergence result to within < 3%. This is consistent with results by Kandhai and Koponen et al. [45, 148] on fibrous media, who found a diameter of 15-20 lattice nodes for the fibre diameter sufficient to achieve resolution-independent results.

In order to use realistic geometries for further pressure drop studies, two porous media from engineering applications were chosen: one sponge-like SiC matrix, and one catalytic converter consisting of a tube filled with spheres.

¹⁷At the time of the publication the author believed that an additional relaxation step on the wall boundary nodes helped further to improve that scheme. Alas, this assumption was never systematically investigated. See also Chap. 6.5.1 'If I did it today ...'.



Figure 4.22: Packed bed of spheres, surface shaded by the pressure.



Figure 4.23: Friction coefficient versus particle diameter.

Geometry Pre-Processing

For both samples, the geometry was digitised using 3D computer tomography (3D-CT). The tomography data were mapped on an equidistant orthogonal mesh for the lattice Boltzmann simulation [51, 145].

Case 1: Catalytic Converter

A cylindrical porous probe with a height of 110 mm and diameter of 80 mm was scanned using 3D-CT with an average resolution of 0.9 mm. This leads to a discretisation of lx * ly * lz = 123 * 90 * 90 voxel. The complex geometry data were centred inside an lx * ly * lz = 250 * 99 * 99 sized channel (see Fig. 4.24), and a flow with a Reynolds number of approximately $Re \approx 0.1$ was simulated using velocity inlet and pressure outlet boundary conditions.



Figure 4.24: Computer tomography data as input for the Lattice-Boltzmann simulation: catalytic converter.

The simulation was performed on one processor of a VPP 700 at the Leibniz-Rechenzentrum in Munich; 50,000 iterations were necessary for this set-up, which took about 25,200 CPU seconds and 850 MBYTE of computer memory were necessary for the storage of the $\approx 2.45 * 10^6$ voxel.



Figure 4.25: Pressure field for the flow through a catalytic converter (x-z plane).

A cylindrical porous probe with a height of 30 mm and a diameter of 82 mm was scanned using 3D-CT with an average resolution of 0.5 mm. This leads to a discretisation of lx * ly * lz = 44 * 147 * 147 voxel (see Fig. 4.26). The average diameter of the flow channels is large enough with this resolution to produce resolution-independent results. The complex geometry data were centered inside an lx * ly * lz = 100 * 149 * 149 sized channel, and a flow for a Reynolds number of about $Re \approx 0.1$ was simulated using velocity inlet and pressure outlet boundary conditions.



Figure 4.26: Computer tomography data as input for the Lattice-Boltzmann simulation: SiC matrix (right: section).

The simulation was performed on one processor of a VPP 700 at the Leibniz-Rechenzentrum in Munich; 10,000 iterations were necessary for this set-up, which took about 5760 CPU seconds¹⁸ and 800 MBYTE of computer memory were necessary for the storage of the $\approx 2.2 * 10^6$ voxel. The pressure distribution in a cross-section and the velocity iso-surface (isotache) can be seen in Fig. 4.27.

Analysis of the Pressure Drop from Experimental Data

As argued above, the tortuosity is obviously not the only reason for the higher pressure drop observed in experiments and numerical simulations when compared to the results derived from the capillaric theories.

The total dissipation in the flow when passing through a porous media can be expressed by:

$$\Phi = -\tau_{ij} \frac{\partial U_j}{\partial x_i} \tag{4.28}$$

¹⁸This simulation was repeated with the current sparse implementation on one CPU of the NEC SX-8 vector computer in less then 500 seconds.



Figure 4.27: Pressure field for the flow through a porous SiC matrix (left: x-z plane, right: isosurface of the flow velocity (isotache), shaded by the pressure).

with

$$-\tau_{ij} = -\mu \left(\frac{\partial U_j}{\partial x_i} + \frac{\partial U_i}{\partial x_j} \right) - \underbrace{\frac{2}{3} \mu \, \delta_{ij} \frac{\partial U_j}{\partial x_i}}_{= 0}$$
(4.29)

for incompressible fluid.

Eqn. 4.28 can be rewritten as

$$\Phi = -\mu \left(\frac{\partial U_j}{\partial x_i} + \frac{\partial U_i}{\partial x_j} \right) \frac{\partial U_j}{\partial x_i}$$

$$= -\mu \left\{ \underbrace{\left[\left(\frac{\partial u_1}{\partial x_1} \right)^2 + \left(\frac{\partial u_2}{\partial x_2} \right)^2 + \left(\frac{\partial u_3}{\partial x_3} \right)^2 \right]}_{+ \underbrace{\left(\frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2} \right)^2 + \left(\frac{\partial u_3}{\partial x_2} + \frac{\partial u_2}{\partial x_3} \right)^2 + \left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} \right)^2}_{\Phi_s \text{ (shear)}} \right\}$$
(4.30)

The dissipation can thus be expressed as a sum of two parts: the dissipation caused by shear forces Φ_s and the dissipation caused by elongational strain Φ_e :

$$\Phi = \Phi_s + \Phi_e . \tag{4.31}$$

An evaluation of the detailed flow fields produced by the numerical simulation with the fraction, Φ_e/Φ_s , yields:

Example 1 (Catalytic Converter):

Friction coefficient	$\Lambda = 217.6$
Dissipation (elongation)	$\Phi_e = 2.111 * 10^{-02}$
Dissipation (shear)	$\Phi_s = 2.854 * 10^{-02}$
elongation / shear	$\Phi_e/\Phi_s=0.74$

Example 2 (SiC matrix):

Friction coefficient	$\Lambda = 342.4$
Dissipation (elongation)	$\Phi_e = 3.853 * 10^{-04}$
Dissipation (shear)	$\Phi_s = 6.486 * 10^{-04}$
elongation / shear	$\Phi_e/\Phi_s=0.59$

In both examples, the friction coefficient Λ is much larger than predicted by the capillaric theory. A considerable amount of the pressure drop is caused by the elongational strain Φ_e , what can be clearly observed from the relation Φ_e/Φ_s .

Similar to the case of a packed bed of spheres presented in the previous section, an explanation of the tortuosity only by the increased length of the flow channels is not correct.

It shall be noted that for both examples the friction coefficient is higher than that of the previous test case. For the catalytic converter, possibly the confining tube and a different (denser) packing of the spheres is responsible for the increase. Also the resolution of the spheres with a diameter of $D_p \approx 10 - 15$ is slightly too coarse, resulting in a too high friction coefficient (see Fig. 4.23). Due to its completely different surface structure, a direct comparison of the friction coefficient of this geometry with the data measured by Durst for a packed bed of spheres is not reasonable.

Case 3: Randomly Distributed Cubes

Geometry and Boundary Conditions For a refined test, five samples with a porosity of $\varepsilon = 0.75$ were generated by randomly distributing cubes of edge length $l_e = 10$ lattice units inside a $l_x * l_y * l_z = 100 * 50 * 50$ domain. Overlapping of the cubes was not allowed in order to control the shape and specific surface of this artificial porous media.¹⁹

The whole set was centred inside a $l_x * l_y * l_z = 200 * 50 * 50$ sized channel and periodic flow boundary conditions were applied orthogonal to the mean flow direction. At the inlet, a constant velocity profile was applied, and at the outlet the pressure was fixed.

Simulation Parameters For each of the five samples, six simulations with Reynolds Numbers in the range $0.01 \le Re \le 100$ were performed on a SGI Enterprise 2000 with an 270 MHZ IP27 Processor. 180 MBYTE of memory were used for the 500,000 lattice nodes, and for 20,000 iterations 45,000 s of CPU-time were consumed.

¹⁹The use of randomly distributed square shapes for Lattice-Gas simulations was first suggested by Kohring [43].

Simulation Results The friction factor can be derived from the flow field with the help of the above mentioned equations. It is known from experimental investigations, that the friction factor is a constant for Re < 1 and increases linearly with the Reynolds number. As can be seen in Fig. 4.28 (left), the experimental results by Durst et al. [14] predicting a relation $\Lambda = 182 + 1.75 * Re$ for the Reynolds number dependent friction factor are approximated very well by the simulation results.



Figure 4.28: Left: friction factor as a function of Reynolds number, right: relation of elongation and shear, both for five different geometries and six different Reynolds numbers.

The dissipation caused by shear and elongation can be determined from the flow field by evaluating Eqn. 4.30. As expressed by Eqn. 4.31, these two parts sum up to the total dissipation and must therefore be directly related to the friction factor. This was investigated in detail for the five simulation results at $Re \approx 0.22$. As can be seen from the plot on the right hand side of Fig. 4.29, Φ is strongly related to Λ .

This result is in good agreement with the observation made in the two previous studies with a catalytic converter and a SiC structure, presented in the previous section.



Figure 4.29: Left: dissipation by shear, elongation and sum, right: total dissipation.

Comparing the tortuosity for the five different geometries (Fig. 4.30 left) with the total dissipation (Fig. 4.29 right), one can see that these two quantities are not related to each other and that the average tortuosity ($\tau \approx 1.12$, measured by integrating the average length of streamlines) is much too small to explain the gap between capillaric theory and numerical



Figure 4.30: Correlation between left: tortuosity, right: relationship of elongation and shear.

simulation. Comparing Fig. 4.30 (left) and Fig. 4.30 (right), there is a strong indication that the tortuosity is related to the relation of dissipation caused by elongation and the dissipation caused by shear. Thus, this relationship Φ_e/Φ_s is, as is the tortuosity, directly related to the geometry for low Reynolds number flow. This is also indicated by the almost constant Φ_e/Φ_s over six orders of magnitude for the Reynolds number (Fig. 4.28, right).

4.4.4 Conclusion

Applying the Lattice-Boltzmann technique for low-Reynolds flow simulations through regularly packed beds of spheres, a digitised SiC matrix and randomly generated geometries, we were able to determine the friction coefficient, the tortuosity and the dissipation caused by shear and elongation of the fluid.

It could be shown by the *a posteriori* analysis of the simulated flow fields, that elongational strain gives an important contribution to the pressure drop. Therefore, the derivation of a tortuosity factor from pressure drop measurements might produce a considerable error, when neglecting the dissipation due to elongational strain. The hypothesis of elongated flow paths cannot explain the gap between the friction factor predicted by the capillaric theory and experimental or numerical results.

We believe that the above study, validating almost 20 years old experimental results obtained by Durst,²⁰ is a very good example to exemplify the power of the Lattice-Boltzmann method for quantitatively accurate simulations, particularly for complex flows.

²⁰Due to their obvious contradiction to the widely accepted tortuosity hypothesis, these experimental results were highly criticised at the time of publication.

Chapter 5

Multi-Physics Extensions

After investigating, in the previous chapter, a variety of complex flow problems with the Lattice-Boltzmann method, the aim of this chapter is to review a few extensions towards multi-physics applications. We speak of multi-physics, when more than 'just flow' has to be considered within the simulation.

Studies carried out by Kaandorp on coral growth [41, 42] and transport and erosion processes by Chopard, Masselot and Dupuis [155, 156, 157] are inspiring examples of multiphysics extensions of the Lattice-Boltzmann method. The complexity of simulation results achieved with these models are in fascinating contrast to their inherent simplicity.

We will demonstrate in the following that the Lattice-Boltzmann method is particularly suited for integrating a variety of concepts for modelling multi-physics. It is the locality of the method and the option of simple geometry representation by the marker-and-cell approach, which makes it possible to define and implement efficient models for the simulation of complex, heterogeneous processes which interact with the transient fluid phase.

In this chapter, heterogeneous chemical reaction, adsorption and resorption in complex geometries and milk/blood clotting simulations are presented.

5.1 Chemical Reaction Modelling¹

In chemical engineering, diffusion and mass transport play an important role. The interplay of mass transport and chemical reactions is of particular importance. An investigation of the coupling of these transport phenomena is only possible when simultaneously solving the Navier-Stokes equations and the convection-diffusion-reaction equation. With usual engineering approaches, both equations are often treated separately on a global, or at least homogenised, scale.

¹The research work presented in this section was performed at the C&C Research Laboratories, NEC Europe Ltd. (St.Augustin, Germany). It was presented at the First International Symposium on Advanced Fluid Information - AFI-2001 (Sendai, Japan, 2001) and the Parallel Computational Fluid Dynamics - ParCFD 2002 (Kansai Science City, Japan, 2002), published in [98, 99]. Extensions of the reasearch presented here was performed by T.Zeiser in co-operation with the author and published in [158, 159, 160, 161].

Here we use the Lattice-Boltzmann method for calculating both, the flow of the carrier fluid and the transport of the passive-scalar reacting chemical species simultaneously, using a real digitised 3D geometry. For the carrier fluid and each species, a separate particle density distribution function with different relaxation times (kinematic viscosity or molecular diffusion coefficients) is used based on the algorithm of Flekkøy [162]. All density distribution functions are coupled via the flow velocity which is determined from the carrier fluid. Feedback of the species distributions on the flow field is neglected, so only passive scalar transport is considered. This scheme is a good approximation for many applications, e.g. environmental processes such as pollutant transport in ground-water flow.

As an example (see Fig. 5.1), we consider a generic surface-catalytic heterogeneous chemical reaction between two species, A and B, of the type

$$A + B \rightleftharpoons C \tag{5.1}$$

with a reaction rate r proportional to the concentrations [A] and [B] of the species

$$r = k[A][B] = \frac{\partial[C]}{\partial t} = -\frac{\partial[A]}{\partial t} = -\frac{\partial[B]}{\partial t}$$
(5.2)

which takes place only on the surface of the porous geometry, a fraction of a digitised SiC matrix structure.



Figure 5.1: Snapshot from a time dependent visualisation of the chemical reaction $A + B \rightleftharpoons C$ (shown are the geometry and iso-surface of the product concentration [C]).

Due to the locality of the Lattice-Boltzmann method, the reaction coefficient k is simply made a space-dependent variable to model the heterogeneous catalytic reaction. Local deactivation of the catalyst can also be implemented in this way. The source term due to the chemical reactions is implemented as an additional part in the Lattice-Boltzmann equation which, accordingly to the differential equation 5.2, modifies the local distribution functions after the relaxation process.

5.2 Nonlinear Adsorption / Desorption²

Adsorption/desorption models describe the transfer of a solvent from the fluid into the solid phase (adsorption) or from the solid into the fluid phase (desorption). In most approaches, details of this process which involves the micro-scale diffusion inside the solid-phase are modelled in a simplified way by a rate equation.

This idea can be realised with a Lattice-Boltzmann multi-physics extension by applying a local rule describing the temporal change of the adsorption rate $\partial s/\partial t$ on the solvent concentration *c* and the adsorbed mass *s* (i.e. the immobile mass deposited per unit volume of the porous media matrix) [163]:

$$\frac{\partial s}{\partial t} = r(k_p c^p - s^q) \tag{5.3}$$

with parameters $k_p > 0$ and exponents p,q fulfilling $p/q \le 1$. At equilibrium, i.e. for $\partial s/\partial t = 0$, this model reduces to a Freundlich isotherm

$$s = kc^n \tag{5.4}$$

with $k = k_p^{1/q}$ and n = p/q.

Such a relationship between s and c frequently describes the adsorption of substances like pesticides, polycyclic aromatic hydrocarbons and heavy metals in soil and aquifer sediments (see e.g. [164],[165], [166]) over several orders of magnitude in concentrations.

5.2.1 Lattice-Boltzmann Extension: Adsorption Model

Using a passive-scalar diffusion scheme for the solvent (e.g. based on the algorithm described by Flekkøy [162]), adsorption is simulated within the Lattice-Boltzmann code by introducing arrays for the adsorbed species concentration s_n link-wise on the surface of the solid phase (see Fig. 5.2).

Each time step, the concentrations of the adsorbed species s_n and the solvent c are locally updated by applying Eqn. 5.3 to calculate the amount of concentration to be transferred from the fluid to the solid phase.

5.2.2 Simulation Results

For (effectively) one-dimensional media, Grundy [167] and Jaekel [168] showed by asymptotic analysis that in the quasi-equilibrium case, after pulse-type injection, the concentrations at a fixed position (breakthrough curves) should approach power laws in time:

²The research work presented in this section was performed together with U.Jaekel at the C&C Research Laboratories, NEC Europe Ltd. (St.Augustin, Germany). It was presented at the Parallel Computational Fluid Dynamics - ParCFD 2002 (Kansai Science City, Japan, 2002) and the IVth IMACS Seminar on Monte Carlo Methods - MCM-2003 (Berlin, Germany, 2003), published in [99]. A later version submitted (but not presented) at the the Third International Conference on Computational Science - ICCS 2003 was published in [100].



Figure 5.2: Link-wise concentration of the adsorbed species s_n at the solid surface, and solvent concentration c in the fluid phase.

$$c \propto t^{-\alpha}$$

$$s \propto t^{-\beta}$$
(5.5)

where the exponents $\alpha = 1/(1-n)$ and $\beta = n/(1-n)$ are determined by the Freundlich exponent *n* alone. This is in contrast to the exponential decay of breakthrough curves observed for chemically inert substances without adsorption.

For the rate equation (Eqn. 5.3), the concentrations cannot always approach quasi-equilibrium. However, asymptotic analysis predicts that the quasi-equilibrium asymptote holds for the case q = 1 [163].

For our simulations we exploited Eqn. 5.3 with factors r = 0.1 and $k_p = 0.001$. We performed simulations for the exponents n = p = 0.5 and n = p = 0.8 for a porous media generated from a 3D-CT scan of a SiC-matrix (see Fig. 5.1).

After establishing steady laminar flow, for a short time a small amount of the solvent species was introduced near the inlet and transported downstream through the porous media by advection-diffusion.

After the peak of the concentration passed the porous media an order of 10^6 more iterations were necessary to achieve asymptotic behaviour, which took several CPU-hours on a single-CPU NEC SX-6i vector-computer.

Fig. 5.3 shows the first 45,000 iterations of the time evolution of the solvent concentration and the adsorbed mass at a surface point inside the porous media in a linear plot for the exponent n = p = 0.8. It can be seen that the the solvent concentration reaches a maximum earlier and decays faster than the adsorbed mass.

Figs. 5.4 show the long term behavior of the two exponents n = p = 0.5 (left) and n = p = 0.8 (right). In both cases, one observes an approach towards the power laws predicted in Eqn. 5.5. More detailed simulations with more iterations are necessary to investigate the long term behaviour and show whether power law behaviour was achieved.

Keeping this in mind, with a regression in the linear part over the last 200,000 iterations the slopes were determined to be t = 1.03 (adsorbed species) and t = 2.05 (solvent species) for the exponent n = p = 0.5 as well as t = 4.36 (adsorbed species) and t = 5.45 (solvent



Figure 5.3: Linear plot of the time evolution (first 45,000 iterations) of the concentration of the adsorbed and dissolved fractions for an exponent n = p = 0.8.



Figure 5.4: Logarithmic plot of the time evolution of the concentration of the adsorbed and dissolved fractions for the exponents n = p = 0.5 (left) and n = p = 0.8 (right).

species) for the exponent n = p = 0.8. These are in good agreement with the predicted t = 1.0 (adsorbed) and t = 2.0 (solvent) for the exponent n = p = 0.5 and of the correct order for the exponent n = p = 0.8, where the predicted slopes are t = 4.0 and t = 5.0 respectively. Due to the slower decay resulting from the exponent n = p = 0.8, one should expect the result to approach the theoretical value more accurately when more than the actual 10^6 iterations are performed.

5.2.3 Visualisation

Using the previously described coupling of our Lattice-Boltzmann code with the RVSLIB routines (see Chap. 3.4.1), we were able to produce movies which provide a detailed insight in the time dependent concentrations of the adsorbed and solvent species.

Fig. 5.5 shows two screen snapshots taken from the movie at two different time-steps. The concentration of the solvent is displayed on a cut in the x-y plane, while the concentration of the adsorbed species colours the surface of the porous media.



Figure 5.5: Screen snapshot of the visualisation (flow from the left to the right). Displayed is the concentration of the solvent (x-y plane) and the adsorbed species (on the surface of the porous media) at two different time steps. One can observe the peak concentration of the solvent travelling downstream through the porous media, followed by the peak concentration of the adsorbed species.

The simulation was performed on a NEC SX-6i as a server and a NEC PowerMate PC as a client, both connected by a local area network.

5.2.4 Conclusion

For a non-linear adsorption/desorption model, the preliminary simulation results were found to indicate good agreement with the theoretical predicted power law model.

Adsorption/desorption is an very good example of utilising the special capabilities of the Lattice-Boltzmann method to model multi-physics, namely the explicit discretisation of the geometry and the simplistic introduction of local rules.

5.3 Concurrent Numerical Simulation of Flow and Clotting³

In this chapter, a novel approach for a concurrent numerical simulation of the unsteady flow within an idealised stenosed artery and a simplified milk clotting process,⁴ based on a residence time model is presented. This model is presented as a first step towards modelling the significantly more complex process of blood clotting.⁵

⁵More recent results with advanced blood clotting models, taking into account processes like platelet activation and enzyme reactions, were presented at the Third International Conference for Mesoscopic Methods

³The research work presented in this section is a first result from the joint research of the C&C Research Laboratories, NEC Europe Ltd. (St.Augustin, Germany) and S.E.Harrison (under the supervision of P.V.Lawford and D.R.Hose) in the Academic Unit of Medical Physics, University of Sheffield (UK). It was presented at the 11th International Conference on Parallel and Distributed Systems - ICPADS 2005 (Fukuoka, Japan, 2005), the Second International Conference for Mesoscopic Methods in Engineering and Science - ICMMES 2005 (Hong Kong, China, 2005 - invited talk) and the 14th International Conference on Discrete Simulation of Fluid Dynamics in Complex Systems - DSFD 2005 (Kyoto, Japan, 2005), published in [75, 77, 169, 170, 171]. The simulation results presented in this section were mainly generated by S.E.Harrison [15] under supervision of the author, who implemented the solidification algorithm.

⁴Milk was used as blood analogue in the experiments performed by Smith [172], since it is considerably cheaper and easier to handle than blood. After enzymatic activation, the clotting behaviour is comparable to that of activated blood. For details see [172].

The aim of this study was not to capture all the complex bio-chemical details of the milk clotting process itself, or discuss its relation to blood clotting. This chapter is intended as a 'proof of concept' for the general possibility and efficiency to simulate clotting with the Lattice-Boltzmann method.

In fact, it can be considered as a particularly well suited example of extending a Lattice-Boltzmann flow solver with a multi-physics model to simulate complex transient flow and mass-transfer processes, which result in changing geometry.⁶

Since the subject of medical physics has been the author's major field of research during the past three years, a brief introduction of the medical background, which motivates the clotting simulations, shall be given.

5.3.1 Medical Background

Cardiovascular disease annually claims the lives of approximately 17 million people worldwide [175]. Atherosclerosis is one particular disease which causes the formation of deposits (plaque) on the inner lining of an artery. Plaque rupture may result in emboli, which in turn may lead to myocardial infarction and ischaemic stroke. A secondary concern is that of flow disturbances associated with disease related narrowing of the vessel lumen (vessel stenosis, see Fig. 5.6).

It is likely that areas of stagnant or recirculating flow will develop downstream of a stenosed artery and if activated blood remains in such a region for a prolonged period of time, thrombosis may occur.

Coagulation can be initiated by shear rates of sufficient magnitude to cause cell lysis and release clotting factors [176, 177, 178, 179, 180]. Following activation, the route to coagulation involves a unique cascade of reactions. Several attempts have been made to model the relevant molecular pathways [181, 182, 183, 184, 185, 186] but these fail to consider realistic flow fields and their development with the growth of the thrombus. A comprehensive understanding of thrombosis requires full consideration of the three entities of Virchow's triad; blood chemistry, vessel wall properties and fluid mechanics.

In this first attempt to model blood clotting, we will not deal with the complex pathway to coagulation, but describe a model which covers the aspects of transient flow simulation and clotting of activated blood based on a simple residence time model. This approach is of course a very crude approximation of the highly complex biochemical processes, but it includes the interaction of a time-dependent flow-field with the varying geometry of a growing clot, which has a major influence on the final shape of the clot itself.

in Engineering and Science - ICMMES 2006 (Hampton, Virginia/USA, 2006 - invited talk) and the 15th Discrete Simulation of Fluid Dynamics conference - DSFD 2006 (Geneva, Switzerland, 2006), published in [173, 174] and accepted for publication in [76]. Since the author of this thesis is not the first author of these recent publications, which are mainly part of the impressive research undertaken by S.E.Harrison in the framework of her PhD-thesis [15] (which was partially supervised by the author), these results will not be presented here.

⁶The author believes that the general approach explained in this chapter can be applied to model a whole class of problems where the interaction of flow and mass-transfer influences the solid fraction of the computational domain.



Figure 5.6: Angiographic image of a stenosed coronary artery. Diameter of parent vessel is approximately 3mm (image courtesy of Dr. Julian Gunn).

An extension of the passive-scalar diffusion model implemented into a sparse Lattice-Boltzmann solver described in Chap. 3.1 was used to estimate the residence time of a fluid. By applying a residence-time based clotting model, the increase of the solid fraction (clotting) was simulated together with the flow field, which had to adapt to the constantly changing boundary conditions.

5.3.2 Lattice-Boltzmann Extension: Aging Model

Assuming that clotting occurs after a certain elapsed period since the 'activation' of milk or blood, the residence time of the activated fluid is the most important variable for the clotting process because it indicates the likelihood of clot formation.

A passive scalar is used as a tracer to estimate the residence time of activated fluid in our model. This tracer is transported by advection-diffusion and a small, constant quantity is injected at every lattice node each time step. The local concentration of the tracer is therefore proportional to the average 'age' of the fluid (blood) which can be used as a threshold parameter within the clotting model.

The diffusion coefficient relates to the amount of mixing between different regions of a fluid and is an important parameter that must be chosen carefully.⁷ Recirculation zones with closed streamlines may be produced, for example, immediately downstream of the stenosis and the only mechanism of transport between these regions and the remaining flow is diffusion.

For the carrier fluid and the tracer, separate particle density distribution functions with different relaxation times (relating to the kinematic viscosity or molecular diffusion coeffi-

⁷In this preliminary study the diffusion coefficient was set to a value which produces clots within a time frame more or less comparable to the experiments. A more detailed study revealed that molecular diffusion alone cannot be sufficient for the amount of mixing which was observed in the experiments. More details to this can be found in the thesis of S.E.Harrison [15].

cient) are used, based on the algorithm of Flekkøy [162]. All density distribution functions are coupled via the flow velocity which is determined from the carrier fluid. Feedback of the species distributions on the flow field is neglected, so only passive scalar transport is considered.

5.3.3 Lattice-Boltzmann Extension: Clotting Model

In this approach we assume that a fluid (milk or blood) is able to clot after a certain elapsed period post activation. The local age of the fluid is determined by the concentration of the passive scalar tracer, as described in the previous section.

When the local concentration of the tracer (which is computed at each time-step) reaches a certain threshold, solidification takes place. Within the Lattice-Boltzmann framework this means a fluid node becomes a solid node and the solid surface boundary condition is applied (see Fig. 5.7).



Figure 5.7: Clotting is modelled by increasing the solid fraction when the concentration c of the tracer species reaches a threshold c_t .

During subsequent iterations the flow field and age distribution adapt to the new geometry, while further clotting on adjacent fluid nodes may occur. This allows for the concurrent simulation of solidification and flow, which is believed to be essential for capturing the complex flow-related clot morphology.

5.3.4 Simulation Results

With our Lattice-Boltzmann implementation, all simulations presented in this section were performed on the NEC SX-6i vector computer which has a peak performance of 8 GFLOP/s. Our code always achieved a sustained performance of more than 50% of the peak performance with about 25 MLUP/s for the flow simulation alone. Although these figures indicate a very good vectorisation, the simulation of 320,000 iterations (2.75 s real time) of the turbulent Re = 550 3D flow, to be presented here, required 40 CPU hours.

Flow through an Idealised Stenosis

Prior to simulations using the aging and clotting models, we required verification that the Lattice-Boltzmann flow solver accurately computes flow through an idealised stenosis.

The stenosis is 5mm long forming a square edged occlusion of 75% by area, situated two diameters downstream of the inlet (see Fig. 5.8).



Figure 5.8: Idealised 75% stenosis geometry.

Throughout these simulations we applied a steady parabolic velocity profile at the inlet and constant pressure at the outlet. The non-Newtonian behaviour of blood was neglected.⁸

A simulation of laminar flow with a Reynolds number of Re = 100 was performed which gave a recirculation length of 29 mm.

For the purpose of validation, a perspex test section was constructed of identical geometry. Flow visualisation was accomplished⁹ at Re = 100 by the injection of a dye stream immediately upstream of the stenosis. This dye became entrained near the wall within the stenosis, therefore demarcating the boundary layer between regions of recirculation and the central jet (Fig. 5.9). These results show similar dimensions of the recirculation zone, although exact determination of the reattachment point is not possible.

The next challenge was simulating turbulent flow at a Reynolds number of Re = 550, which is approximately the average Reynolds number within the femoral artery. Achieving turbulent flow at such low Reynolds numbers is not trivial since the flow will tend to damp back into the unsteady laminar regime.

It transpired that the slightly rough walls (generated by randomly varying the radius ± 1 lattice node) we used (lx * ly * lz = 1214 * 98 * 98) in the three dimensional simulations were sufficient to induce and maintain turbulence. After a long transient initial period where complex vortices developed, laminar flow broke down and fully turbulent flow was observed (see Fig. 5.10), remaining stable until the end of the simulation (320,000 iterations).

Qualitatively comparing the computational flow solutions to experimental results, good agreement can be observed between the size and speed of the flow features (see Fig. 5.11).

⁸For non-Newtonian blood flow simulation see e.g. [187], a Carreau-Yasuda model was recently also implemented into the sparse Lattice-Boltzmann solver of the *International Lattice Boltzmann Software Development Consortium*. First simulation results (briefly reviewed in Chap. 6.3) indicate that under certain circumstances the non-Newtonian behaviour of blood flow cannot be neglected.

⁹The experiments were performed by Smith [172].


Figure 5.9: Comparison of experimental (above, from [172]) and numerical (below) results for flow through a 75% stenosis at Re = 100.



Figure 5.10: Snapshot of a velocity iso surface of fully developed turbulent flow through a 75% stenosis at Re = 550.

Aging Simulation

For the previously described aging model the most simple validation involves simulating the age distribution of a laminar fluid in a 2D channel. For this purpose a channel of lx * ly = 200 * 42 lattice nodes was created and allowed to develop a steady, laminar flow profile. At each subsequent iteration, a small amount of the previously discussed passive-scalar tracer was added to the local concentration at all lattice points.

Flow velocity is maximum in the centre of the channel, falling parabolically to zero at the walls. Therefore, after a certain time has elapsed, the 'youngest' fluid will be found in the centre at the inlet and the 'oldest' fluid will be found near the outlet walls.



Figure 5.11: Comparison of simulated (left) and experimental (right) flow through a 75% stenosis at Re = 550 at four different times.



Figure 5.12: Tracer concentration ('age') for parabolic channel flow (dark colour is referring to higher concentrations representing 'elder' fluid).

In previous milk clotting experiments carried out by Smith [172], clot deposition was observed to be maximal in regions distal to the stenosis. One explanation for this is that the recirculation regions retain clottable fluid, allowing it to age and adhere.

2D simulations of laminar (Re = 100) and turbulent (Re = 550) channel flow containing a stenosis of 50% by diameter (equivalent to the 75% by area stenosis in 3D) clearly indicate that the regions with the highest concentration of the tracer, and hence the oldest fluid, are similar in location to the clot depositions produced experimentally.

Clotting Simulation

For the 2D clotting simulation a lattice size of lx * ly = 532 * 82 nodes was used and an initial 200,000 iterations were performed to establish time-dependent flow at Re = 550. Following this, the tracer was injected at a constant rate.

Defining a threshold for the tracer concentration, indicating the age of the fluid, allows us to implement the solidification process: all fluid lattice nodes where a concentration above this threshold is found are solidified and no further mass transport is allowed (see Fig. 5.7). The threshold concentration was chosen to be small enough to allow clotting within a reasonable simulation time and large enough to avoid solidification of many lattice nodes within a few iterations. Obstruction of the outlet due to clot growing from the walls must also be avoided.¹⁰

A further 300,000 iterations (equivalent to 11.7 s in real time) were performed to allow a clot to grow (see Fig. 5.13).



Figure 5.13: Growing clot (black) downstream of a 2D stenosis at different time steps t. The age of the fluid is shown in grey, darker regions indicating older fluid.

In this first approach we exploit the fact that a scale separation exists between the typical rate at which clot forms and the frequency with which flow patterns change.¹¹ This means it is not necessary to adhere to the experimental time scales where a clot grows within several minutes. For the results presented here the typical time-scale associated with clot growth is approximately two orders of magnitude greater than that associated with vortex shedding.

As can be seen in Fig. 5.13, clot growth initiates in the recirculation domain downstream of the stenosis. The size of the clot increases gradually with time, whilst the flow field adapts to the new geometry. Of particular interest is the downstream migration of the recirculation region. Due to vortex shedding, a secondary vortex is established leading to a second

¹⁰Within this preliminary study, the threshold parameter was not derived from physical considerations but simply adjusted to qualitatively reproduce the experimentally observed phenomena.

¹¹A more sophisticated approach making use of a scale separation can be obtained within multi-scale models. Separated time and length-scales are identified on a scale-map [188, 189] and dedicated software tools for each scale are coupled to exchange the required information. Such models are being investigated within the European research project COAST, to which the author contributes with the sparse Lattice-Boltzmann flow solver. A short overview will be given in Chap. 6.4.

concentration maximum approximately one vortex diameter downstream of the clot (see Fig. 5.14).

A secondary clot has been identified experimentally at this Reynolds number (Re = 550), though the relation between this and a secondary vortex must be investigated further.



Figure 5.14: Secondary milk clot (above) in the experiment and secondary peak in the tracer concentration (below: numerical simulation, darker regions indicating older fluid.

The final asymmetric shape of the clot reflects the effect of unsteady flow on the pattern formation procedure, showing some qualitative similarities with results of milk clotting experiments [172] using comparable flows.

A comparison of a Lattice-Boltzmann clotting simulation with experimental results produced by Smith [172] show good qualitative agreement (see Fig. 5.15).



Figure 5.15: Comparison of an experimentally produced milk clot (above) with the Lattice-Boltzmann simulation (below). The position of the stenosis is indicated.

To demonstrate that the computationally demanding extension to three dimensions is feasible, a 3D clotting simulation at Re = 100 was carried out within the stenosis geometry (Fig. 5.8). The shape of the clot (see Fig. 5.16) looks similar to those from the milk clotting experiments, and no artifacts, as e.g. a dependency from the lattice symmetry, were observed.



Figure 5.16: 3D milk clot downstream of a stenosis at Re = 100.

5.3.5 Conclusion

The above study presents a new approach for simulating clotting using the Lattice-Boltzmann technique by using a passive scalar as a tracer of the age of activated fluid. This can be considered as a typical example of a multi-physics extension. The possibility of locally modifying the geometry during the simulation allowed for the interaction of flow and clotting, which is believed to be the key for reproducing the experimental clot shapes.

This section has been restricted to briefly introducing the method. Recent studies (not presented in this thesis) which have been performed at the University of Sheffield in cooperation with the author (for details see [173] and the thesis of S.E.Harrison [15]) consider more complex clotting algorithms for 3D simulations. There it was shown that taking account of the vicinity of a growing clot and flow parameters such as the wall shear stress considerably affects the onset and development of clotting.

Chapter 6

Discussion, Vision and Outlook

In the previous chapters a selection of complex-flow applications and multi-physics extensions of the Lattice-Boltzmann method have been presented. Considering these successful applications of the method (and many more described in the literature), the author believes that Lattice-Boltzmann is not a tool which will one day completely replace the Navier-Stokes based Finite-Element or Finite-Volume techniques, but has clear advantages if applied to suitable problems. The goal of comparing the different approaches should not be to identify 'the best' method, but to find out which method is more suitable for the underlying class of problems.¹

Succi described in his book [13] four general classes concerning the applicability of Lattice-Boltzmann:

- *Don't Use:* Problems which require strong compressibility and substantial heat transfer.
- *Can Use:* These are basically all standard-CFD problems. It is often a question of taste which method to prefer. Alas, the orders of magnitude less person-years invested so far in the development of Lattice-Boltzmann techniques (as turbulence modelling and acceleration schemes) and software packages (including pre and post-processing tools) often set practical limitations. Reaching full maturity is not a conceptual problem per se, but just a matter of time and money invested in developments.
- *Should Use:* Succi identifies here single and multi-phase flows in complex geometries; we believe this was supported (for the single-phase flows) by the examples given in Chap. 4.
- *Must Use:* This class was described by Succi as 'rather dilute'. In contrast to this statement from the year 2002, we believe, by the examples given in the previous Chapters (4 and 5) that it became obvious that the 'Must Use' class can be populated

¹It might well be argued that some classical CFD-benchmarks the Lattice-Boltzmann method was involved in the past are not very suitable for showing the advantage of the method. Even today there is a certain tendency to apply the method in areas where conventional, well established software packages have certain advantages.

by a variety of complex-flow multi-physics applications. For simulating the interaction of complex fluid flow, mass transport and a related modification of the flow domain, adequate modelling in the framework of traditional Navier-Stokes methods would be incredibly complicated and numerically expensive, if possible at all. The specific way in which solid boundaries are handled within the Lattice-Boltzmann algorithm, together with the simplistic cellular automata-like option of implementing simple local rules, from which complex physical phenomena emerge on larger time and spatial scales, is a clear advantage of this method.

Another new, and hardly explored area, where the focus is on the fast simulation of flow phenomena as opposed to precision of the results, is real-time CFD simulation on the computer's graphics card (GPU) for computer games and other related fields (see Chap. 3.3.2). A very recent impressive example in this area is the MoXicode [90], a real-time simulation of ink dispersion in absorbent paper on a GPU. Using a digital pencil on a touch pad, the user can see the ink 'flowing' into the paper when drawing his calligraphy.

To further populate the *Should Use* and *Must Use* class of Lattice-Boltzmann applications, the author believes it is time for the Lattice-Boltzmann method and its researchers to become involved to do 'the real thing'. That is, particularly:

- Identify further areas where application of the Lattice-Boltzmann method is beneficial. Focus the research on producing quantitatively validated results within competitive simulation times, achieved by performance oriented implementations.
- Leave the small pond of the community, meeting twice a year at the ICMMES and DSFD international conference, and face the challenge of presenting the results at established CFD-conferences.
- Learn from the two generations' experience of classical CFD and port suitable schemes such as multi-grid² and a variety of turbulence models³
- Involve Lattice-Boltzmann in projects (industrial and research) where results are required in a reasonable time within a production environment.

Considering these statements, the author hopes his research of the past ten years was contributing to the first two points. While point three should be challenged by those researchers with a focus on theoretical work, the last point is more and more becoming the focal point of the author's work. Lattice-Boltzmann, though beautiful, is not important enough to justify theoretical research accompanied by a few illustrative applications. The justification of all research in this field, in the end, is a successful application. That is, the solution of real-world problems.

²One approach towards Lattice-Boltzmann multigrid was done by Tölke et al. [190].

³Subgrid models for Lattice-Boltzmann are meanwhile well established (see e.g. [132, 191, 192, 193, 110]). This is not (yet) the case for $k - \varepsilon$ and other two-equation models.

Guided by the paradigm of practicality, a short description of possible application strategies of the Lattice-Boltzmann method will be presented after a few comments related to software development. Arguments for both, the researchers and industry, are given to illustrate why an application of the method is of mutual benefit.

The final two sections briefly introduce two exciting research projects in which the Lattice-Boltzmann method provides the flow solver. An application in these areas – complex-flow medical physics and multi-scale multi-physics – is certainly a field where the method is suitable. We are confident the future will prove that these kinds of applications are located somewhere between the 'Should Use' and 'Must Use' class.

6.1 Software Development

In contrast to the rapid development in theoretical and academic research, unfortunately, *so far* there is only one commercial Lattice-Gas/Boltzmann software package, PowerFLOW by EXA Corporation (founded in 1991 by Kim Molvig). EXA has been very successful particularly in automobile aerodynamics, although the company is active in a range of industries, including automotive, aerospace, engineering, architectural, environmental, and government. EXA currently has 100 employees worldwide, and its customer list contains Audi, BMW, DaimlerChrysler, Dodge Motorsports, Fiat, Ford, Hyundai, Jaguar, Nissan, Paccar, Porsche, Renault, Toyota, Volkswagen and many others.⁴

What made the EXA code so successful? It is debatable whether it was really the superiority of their specific approach in Lattice-Gas/Boltzmann over traditional CFD. Certainly, it was EXA's clear understanding of providing the user with a simple and robust method, an easy to handle user interface which makes tedious meshing unnecessary and an integrated post-processing solution. It was the understanding of the EXA managers that the development of a new commercial CFD software requires sufficient investment in manpower.

Aside from EXA, not a single company or investor was bold enough to spend sufficient money for a strong team of developers. There are a few exceptions where one or two researchers are paid by industry for carrying out specific developments with some good success,⁵ but in general the method is still an academic research topic with little impact on the market.

Besides the most popular 'one (PhD) student one code' approach, a few promising initiatives for sustained software development have emerged:

• The ParPac-code developed at ITWM Kaiserslautern is coupled with the MAGMA-SOFT package [195], the standard simulation and visualisation software for casting processes. The code is able to capture free surface flow and was recently extended with a Bingham-model.

⁴Information taken from Exa's homepage [194].

⁵The author is only aware of Shell Research, the ITWM Kaiserslautern and the IT Research Division of the NEC Laboratories Europe.

- The OpenLB-software, established as an open-source project under the GPL by J.Latt (University of Geneva), V.Heuveline and M.Krause (Super-computing center, University of Karlsruhe). The software is implemented in C++ and some basic flow studies (flow past a cylinder and backward facing step) are presented on the project home-page [196].
- The International Lattice-Boltzmann Software Development Consortium, a 'restricted open-source' project initiated by the author and supported by the NEC Laboratories Europe. The focus of this consortium is to develop a high-performance oriented Lattice-Boltzmann software package for advanced research and production simulations. The recent MPI implementation achieved an update rate of 3.7×10^9 lattice site updates per second on 128 CPUs of the NEC SX-8 vector-supercomputer at HLRS Stuttgart for a medical flow problem (see Chap. 3.2).

The Deutsche Forschungsgemeinschaft (DFG) financed a long-term project called 'Lattice-Boltzmann Arbeitsgruppe' with the aim to perform benchmarks of Lattice-Boltzmann and Navier-Stokes solvers.

Recently the European Commission decided to support the development of the method and its integration into a complex tool-chain in the area of medical physics in the two projects COAST and @neurIST (see Chap. 6.3 and 6.4), where the author has the pleasure to contribute.

Software engineering and maintenance, even of large research or production codes, is still often considered as a task that students can do without specific education and support along-side their research work.

The author can confirm from his own experience in co-ordinating the development of two large Lattice-Boltzmann packages, *BEST*, and since 2003, the software developed in the framework of the *International Lattice-Boltzmann Software Development Consortium*, that performing a sustained development – starting form the basic software design, implementation and benchmarking of prototypes to co-ordinating the steady input of various researchers to a rapidly growing code – is an art on its own, which requires a variety of skills and lots of experience *and time*.

Real progress in the area of Lattice-Boltzmann can – with regards to the currently available advanced techniques – only be achieved, if teaching and application of the required software engineering skills is conducted with the same priority as the method development itself, preferably in a team providing knowledge transfer from senior developers to the younger students.

6.2 Perspectives of the Lattice Boltzmann Method for Industrial Applications⁶

6.2.1 Application Strategies

The reduction of turn-around times in the design cycle is an important aspect of product development in many different industries. New tools for efficient numerical simulation on HPC systems therefore play an increasingly important role in the simulation of fluid- and aerodynamics, e.g., in the areas of automobile design and chemical engineering.

The Lattice-Boltzmann method has two outstanding features, when compared to other conventional CFD methods, which make it particularly suitable for addressing these challenges:

- Very high performance (usually more than 50 % of the peak performance) and nearly ideal scalability on high performance vector-parallel computers (see Chap. 3.1).
- Very efficient handling of the equidistant Cartesian meshes, which are generated by the semi-automatic discretisation of arbitrarily complex geometries (conversion of digital images, 3D CT data or CAD data, see Chap. 2.2.4).

During the first ten years of its development, the Lattice-Boltzmann method was almost exclusively applied for academic research purposes, but a significant tendency towards industrial applications can now be observed.

In the following, we will discuss the advantages of an industrial application of the Lattice-Boltzmann method for the two parties potentially involved: in industry, the LB methods are not yet widely known as a potential alternative to classical CFD methods, whereas some LB researchers apparently do not yet see the advantages of an industrial application of their research work.

6.2.2 Benefits of Industrial Simulations for the Lattice-Boltzmann Community...

A fact which is sometimes ignored by purely academic researchers, is the possible improvement of their numerical method and software, initiated or required by the application in production environments. The feedback of the quality of the simulation results, based on the extensive experience and/or databases compiled by industrial engineers, is a good indicator of the reliability of the method. Time-constraints concerning the availability of expensive hardware and tight schedules require an efficient and robust implementation.

Driven by the demand of continuous improvement, suggestions for further developments can be given by the industrial engineers. This can lead to a fruitful cooperation with a

⁶The concept behind this section was developed at the C&C Research Laboratories, NEC Europe Ltd. (St.Augustin, Germany). It was presented at the Parallel Computational Fluid Dynamics - ParCFD 2000 (Trondheim, Norway, 2000) and published in [197].

mutual exchange of experience, resulting in a fast and efficient development of the method or its implementation.

Usually in industry, different numerical codes are in use or under investigation. Participation in such an evaluation program is a good opportunity for finding out the advantages and disadvantages of the Lattice-Boltzmann method when compared with other commercial Navier-Stokes based codes.

6.2.3 ... and for the Industry

In companies where HPC platforms are already installed, large scale simulations need to be performed with software making optimal use of these expensive and powerful machines. As demonstrated in Chap. 3.1, Lattice-Boltzmann codes can be implemented almost optimally for high end vector-parallel platforms.

Lattice-Boltzmann codes are typically characterised by a relatively easy integration of complex boundaries. The underlying scheme for geometry discretisation allows a straightforward integration of arbitrary complex geometries, which can either be derived from CAD data by special software, or by 3D CT (see Chap. 2.2.4). By providing a user interface, it is not necessary to have a highly specialised CFD expert generating the mesh and running the simulation, a procedure which might easily take several weeks for complicated geometries. The relative simplicity of conducting simulations with the Lattice-Boltzmann method can lead to a significant cost reduction during the industrial design process, and the simulation results are usually available within short time.

Areas where CFD normally fails, due to the impossibility of efficient mesh generation for complex geometries, or the complexity of the physics to be modelled (e.g., simulation of heterogeneous catalytic reactions in chemical engineering) are also potential candidates for the Lattice-Boltzmann method. The simple marker and cell approach, or more advanced schemes, in combination with 3D CT allow the discretisation of almost every geometry and the simulation of domains containing several 10^8 lattice nodes on HPC platforms.

6.3 Blood Flow Simulation in Cerebral Aneurysms: A Lattice-Boltzmann Medical Physics Application within the @neurIST Project⁷

Computer simulations play an increasingly important role in the area of medical physics, from fundamental research to patient specific treatment planning. A potential application

⁷The research work presented in this section is performed together with D.Wang and G.Berti at the IT Research Division of the NEC Laboratories Europe, NEC Europe Ltd. (St.Augustin, Germany) and was presented at the 4th International Intracranial Stent Meeting - ICS 2007 (Kyoto, Japan, 2007), the Fourth International Conference for Mesoscopic Methods in Engineering and Science - ICMMES 2007 (Munich, Germany, 2007 - invited talk) and the Parallel Computational Fluid Dynamics - ParCFD 2007 (Antalya, Turkey, 2007), submitted for publication in [198, 199, 200]. Financial support of the European project @neurIST (contract no. IST-027703) is gratefully acknowledged. Public information about the project can be found on the project web-page [201].

of numerical blood flow simulation is to aid decision making processes during treatment of cardiovascular disease. One example of this is the treatment of aneurysms. Aneurysms are extreme widenings of vessels which can be, if they rupture, life threatening.

A large European project on rupture-risk assessment of cerebral aneurysm – @neurIST – is funded by the European commission for a period of four years. Researchers from a variety of fields collaborate to integrate data from medical studies and simulation tools into a large framework, with the goal to provide clinicians with a decision-support tool on hand.

One particular application we address in the @neurIST framework, is the simulation of blood flow in cerebral aneurysms in domains created from medical images. Our focus is on the efficient implementation of the Lattice-Boltzmann method for this type of medical application as well as considering the correct blood rheology.

Previous studies with medical geometries are reported in [202]. Artoli describes the transient flow reconstructed from an MRI data (further discussion of the background and literature can be found in his thesis [202, pp. 91 ff.]). Alas, the non-Newtonian blood rheology is not considered in this particular simulation.

Beronov [203] applied the concept developed by Artoli for flow simulation in a cerebral aneurysm. The simulations are performed directly on the mesh from the MRI device, and no reconstruction of the surface mesh or fit of the resolution to the requirements of the numerical tool, was accomplished. Also, rheological aspects are neglected, and only a steady inflow with a very low Reynolds number is considered.

6.3.1 Medical Background

One method of treatment of cerebral aneurysms involves insertion of a metal frame known as a stent, to divert flow from the aneurysm. An alternative is to pack the aneurysm with wire; a procedure known as coiling. The resulting modification of the flow field triggers the process of blood clotting (see Chap. 5.3) inside the aneurysm and in future, the flow-field following treatment can be predicted by computer simulation. This may ultimately give an insight into the success of the treatment and long-term prognosis. *In vivo* measurements of specific flow properties are possible, but usually not precise enough to predict for example, wall shear stress or pressure distribution with a sufficient spatial resolution. Since invasive treatments of the brain can be problematic, a pre-surgery risk assessment for the likelihood of rupture of the aneurysm in question is a challenging goal.

To achieve this goal, necessary steps for an accurate numerical simulation of flow properties within an untreated aneurysm together with preliminary results will be briefly introduced in the following.

6.3.2 Image Segmentation⁸

Discretising the geometry for flow simulations from CT or MR data is a challenging task. Depending on the applied method, the resulting geometry can vary, and advanced methods must be applied to generate suitable meshes. This may include manipulations of the triangulated surface mesh representing the geometry (Fig. 6.1 left). Usually, from these data, in the case of Lattice-Boltzmann, a voxel mesh with adequate resolution is generated (Fig. 6.1 right).



Figure 6.1: Left: Tetrahedral mesh of a cerebral aneurysm (courtesy of the @neurIST Project). Right: voxel mesh of the Lattice-Boltzmann simulation.

6.3.3 Non-Newtonian Model⁹

The literature on blood rheology gives a strong indication that the non-Newtonian effects of blood flow may not be neglected for a variety of geometries (see, e.g. [204, 205, 206]). Particularly when estimating the rupture-risk within cerebral aneurysms, the precise knowledge of quantities such as pressure distribution and wall shear stress are expected to be crucial.

Normally, the Casson's model, the power law model and the Carreau-Yasuda (C-Y) model are used to simulate shear thinning blood flow [207]. However, the Casson's model is only valid over a small range of shear rates and the determination of yield stress in the equation is questionable, while the disadvantages of the power law model are the high gradient and potentially infinite viscosity. The C-Y model can overcome these shortcomings and can therefore be considered as the most suitable. It is written as:

$$\frac{\mu - \mu_{\infty}}{\mu_0 - \mu_{\infty}} = (1 + (\lambda \dot{\gamma})^a)^{(n-1)/a}$$
(6.1)

⁸The surface mesh of the cerebral aneurysm was generated by A.Marzo (University of Sheffield) and A.Radaelli (Universitat Pompeu Fabra, Barcelona). The voxel mesh for the Lattice-Boltzmann simulation was produced by G.Berti (NEC Laboratories Europe).

⁹The Carreau-Yasuda model was implemented by D.Wang, supported by the author. Further support of P.Lammers (HLRS) for improving the implementation is gratefully acknowledged.

where μ_0 and μ_{∞} are the dynamic viscosities at zero and infinite shear rate respectively, $\dot{\gamma}$ is the shear rate and λ is a characteristic viscoelastic time of the fluid. At the critical shear rate $1/\lambda$ the viscosity begins to decrease. The power law index parameters *a* and *n* can be determined from experimental data. In our simulations we apply the following set of parameters for a blood analogue fluid [204]: $\mu_0 = 0.022$ Pa s, $\mu_{\infty} = 0.0022$ Pa s, a = 0.644, n = 0.392, $\lambda = 0.110$ s.

6.3.4 Flow Simulation

The MRI patient data were segmented and post-processed on a triangular mesh (see Fig. 6.1 left). From these data, a Lattice-Boltzmann voxel mesh (see Fig. 6.1 right) of size lx * ly * lz = 160 * 120 * 100 was generated, which is of sufficient resolution to obtain mesh-converged simulation. The computational domain contains 1.58 million fluid nodes.

10,000 iterations were performed to reach a converged result, which required 88 s CPUtime on one CPU of CCRLE's NEC SX-8 vector-computer. This is equivalent to an update rate of 39 MLUP/s.

Flow Parameters

For a preliminary simulation, flow at a low Reynolds number of Re = 20 was considered, and a direct comparison of Newtonian and non-Newtonian flow was performed. In order to investigate the influence of the Reynolds number on the non-Newtonian effect, a second simulation comparing the wall shear stress for two different Reynolds numbers, Re = 1 and Re = 120 was performed.

The parameters of the C-Y model were chosen to allow the non-Newtonian blood rheology to be captured. Since a definition of Reynolds number is difficult for non-Newtonian flow, it was ensured that the total pressure loss was equivalent to that of the Newtonian simulation for the same average cross-section velocity.

At the inlet, a constant flow velocity was applied and the pressure at the outlet was kept constant. Bounce-back wall boundaries were applied resulting in zero flow velocity at the wall.

6.3.5 Simulation Results

The streamlines in Fig. 6.2 reveal a complex flow pattern within the aneurysm. It can also be seen that only a small portion of the flow enters the aneurysm, while the majority of trajectories directly follow the main branch (this is due to the specific geometry and the very low Reynolds number of Re = 20). The vortex-like structure inside the aneurysm triggered by the main flow in the artery can be clearly identified.

As can be seen by direct comparison of Newtonian and non-Newtonian simulation results in Fig. 6.3, the shear stress distribution plotted at a cutting plane inside the aneurysm differs significantly: the non-Newtonian simulation produces a larger region with higher shear rates inside the aneurysm.



Figure 6.2: Streamlines indicating the complex flow inside the cerebral aneurysm.



Figure 6.3: Shear stress distribution at an x-z cutting plane inside the cerebral aneurysm. Left: Newtonian, right: non-Newtonian.

For both models, Fig. 6.4 compares the peak of the wall shear stress distribution near the neck of the aneurysm. It can be observed that the maximum value is underestimated, if non-Newtonian effects are not taken into account.

Similar to the previous study at Re = 20, also for the even lower Reynolds number Re = 1 the wall shear stress at the neck of the aneurysm is much higher when the non-Newtonian model is applied (see Fig. 6.5).

These preliminary results indicate that non-Newtonian effects might have a significant contribution for low-Reynolds number flow through cerebral aneurysm.

In a next step, the comparison between the Newtonian and non-Newtonian model was performed for higher Reynolds numbers. From Eqn. 6.1 can be concluded, that for the higher shear rates present at higher Reynolds numbers, the viscosity μ approaches the upper limit of μ_{∞} . In this case, the difference between a Newtonian and a non-Newtonian





Figure 6.4: Wall shear stress the aneurysm (darker colours in Left: Newtonian, right: non-Newtonian.

distribution near the neck of indicate higher shear stress).

simulation will vanish. This assumption is confirmed by the simulation results (see Fig. 6.6) when increasing the Reynolds number to a value of Re = 120.



Figure 6.5: Wall shear stress distribution at Re = 1 (darker colours indicate higher shear stress). Left: Newtonian, right: non-Newtonian.

Since Reynolds numbers in intra-cranial arteries can be as high as Re = 600 - 700 [208], the above results indicate that within the cardiac cycle flow regimes are reached where the non-Newtonian effect vanishes. Currently, it is an open question whether or not non-Newtonian effects have to be taken into account for the blood flow simulation within cerebral aneurysms.

A critical indicator for the growth process and rupture risk of a cerebral aneurysm is the oscillatory wall shear stress [209]. Within this ongoing study we will analyse to which extent the differences of the wall shear stress observed at lower Reynolds numbers contribute to the oscillatory wall shear stress within a full cardiac cycle. This will help to answer the question if non-Newtonian models are required for blood flow simulations in cerebral aneurysm.



Figure 6.6: Wall shear stress distribution at Re = 120 (darker colours indicate higher shear stress, a different colour scale compared to the above images was used). Left: Newtonian, right: non-Newtonian.

6.4 Multi-Scale Multi-Science Simulation: A Lattice-Boltzmann Medical Physics Application within the COAST Project¹⁰

Much smaller than the Integrated Project (IP) @neurIST is the Information Society Technologies (IST) project COAST. Five groups with expertise in the areas of Cellular-Automata, Lattice-Boltzmann, High-Performance-Computing and medical physics combine forces to develop a new framework for multi-scale, multi-science simulations. The COAST application is the modelling of a specific complication which may occur during the treatment of coronary artery diseases, the so called in-stent re-stenosis.

The use of a stent together with balloon angioplasty is a common method of re-opening a stenosed vessel lumen and modelling can be used to predict the post-treatment blood flow field. However, the stent material can induce blood clotting and lead to in-stent re-stenosis, which is an unwanted post-treatment narrowing of the vessel lumen. Specially coated stents can help prevent this effect. In terms of CFD, this is a complex-flow, multi-physics problem with chemical/biological processes on a variety of time and length-scales.

A complex scheme in terms of a scalable hierarchical aggregation of Cellular-Automata and agent-based models with appropriate couplings has to be implemented to address this problem. Simply said, the idea behind this approach is not to solve all scales of the problem (reaching from milliseconds to months and micrometers to meters) within one piece of software, but to identify time and space disjoint entities which can be addressed by specific implementations. These entities then have to be coupled for information exchange in a suitable way. A selection of multi-scale Lattice-Boltzmann simulations are described in [188].

¹⁰The research work presented in this section is performed together with D.Wang at the IT Research Division of the NEC Laboratories Europe, NEC Europe Ltd. (St.Augustin, Germany). Financial support of the European project COAST (contract no. 033664) is gratefully acknowledged. Public information about the project can be found on the project web-page [210].

This ambitious project took off in September 2006 with the definition of a scale-map (see Fig. 6.7) to identify the disjoint time and space scales of all relevant processes involved in in-stent re-stenosis.



Figure 6.7: Scale-map for in-stent re-stenosis, image taken from [189].

The sparse Lattice-Boltzmann flow solver developed within the *International Lattice-Boltzmann Software Development Consortium* will be applied to solve the bulk-flow. The challenge in the framework of this project is two-fold: firstly, known methods for local mesh refinement and higher order boundary conditions have to be implemented in a high performance production code. Secondly, the coupling to the software addressing the time and length-scale of species transport to the endothelium (the first layer of cells of the artery wall which is in contact with the fluid), most likely a particle-based method taking explicitly into account the shape and movement of individual red blood cells, has to be achieved.

6.5 Concluding Remarks

This thesis is a retrospective view on ten years of research in the areas of fluid dynamics, chemical engineering and medical physics. The research question behind this work, highlighting the Lattice-Boltzmann method as a powerful tool to address complex-flow multiphysics problems, was answered by a selection of mostly quantitatively validated studies. These studies spawned from classical CFD problems via advanced modelling of adsorption until very recent applications of blood-flow and clotting simulations, which required the development of several new extensions to the standard flow solver.

6.5.1 If I did it Today ...

Although all these results are still valid today, the numerical method which was used to produce them experienced a considerable development through the past ten years, thanks to the very active Lattice-Boltzmann community.

Several simulations presented in this thesis, performed with what was at their time a state of the art implementation, can nowadays be accomplished in a more efficient or more elegant way. It is beyond the scope of this thesis to reproduce all the results presented in Chap. 4 and 5 with the latest version of our code.¹¹ Instead, it shall be briefly sketched which methods can and should be applied, if these simulations were performed today.

Channel Flow in Increasingly Complex Geometry (Chap. 4.1)

This study, using a geometry made up of an increasing number of square boxes with a long inlet and outlet region, could significantly benefit from local mesh refinement, providing a finer resolution around the square shape obstacles and a coarse resolution in the inlet and outlet region. Also a sparse implementation (see Chap. 3.1.1) would help to safe memory which was allocated also for the occupied lattice nodes.

Decay of Turbulence (Chap. 4.2)

In this study, the time evolution of a vortex and the decay of an isotropic field of turbulence was investigated. A problem here was the initialisation of the density distribution function from the given velocity field. In our approach, we initialised with an equilibrium distribution function and let the system develop for a few iterations before taking the first measurement as reference at t = 0. This does not lead to wrong results, but to accomplish a start from exactly the given velocity field, it would be a better idea to follow a suggestion from Luo [211] and keep the velocity fixed while letting the system develop for a short time.

Plain Channel Flow around a Square Cylinder (Chap. 4.3)

The transient flow around a square obstacle mounted in a channel was investigated in this section. As in the above example from Chap. 4, local mesh refinement and a sparse implementation would help to significantly improve the performance and save computer memory. Since the process of vortex shedding is critically influenced by the details of the development of the vortices on the surface of the square obstacle, local mesh refinement is expected to be particularly beneficial, because it allows a very fine resolution of this region, while keeping the overall number of grid points small. A sufficiently accurate capturing of the vortex shedding process has influence on the accuracy of measured quantities as Strouhal number as well as lift and drag. Discrepancies between the Finite-Volume and Lattice-Boltzmann results concerning lift and drag at higher Reynolds numbers can be explained by an insufficient resolution of the equidistant Lattice-Boltzmann mesh around the square obstacle, which could be cured by applying local mesh refinement.

¹¹Which would not change the results, but only help to produce them with less computational effort.

Numerical Analysis of the Pressure Drop in Porous Media Flow (Chap. 4.4)

Porous media flow is one of the domains where the Lattice-Boltzmann method is particularly advantageous. The problem that the wall position for the BGK scheme in combination with standard bounce-back wall boundary condition is critically dependent on the relaxation parameter was discussed in Chap. 2.2.3. Further, an approximation of curved surfaces by the voxel mesh might lead to additional errors.

In the porous media flow studies presented in Chap. 4.4 we tried to avoid these problems by first measuring the accuracy of approximating a circular shape with square obstacles and then calibrating the correct relaxation parameter, which was kept fixed throughout the simulations. Although a tedious procedure requiring a high resolution for the curved objects and giving away the option of freely adjusting the viscosity via the relaxation parameter, correct results could be produced with the applied Lattice-Boltzmann scheme.

Recent (not yet published) studies investigating the flow through porous structures for Diesel particle filters¹² proved the advantage of the new TRT and MRT schemes (instead of BGK), which keep the position of the solid-fluid interface fixed over a wide range of relaxation parameters.

The additional use of improved wall boundary conditions for the study on spherical obstacles could further help to reduce the required resolution, particularly in combination with local mesh refinement.

Similar to the previous studies containing a certain amount of solid fraction, a sparse implementation would help to save memory.

Nonlinear Adsorption / Desorption (Chap. 5.2)

In this chapter an approach for nonlinear adsorption and desorption modelling was presented. Although the model was relatively simple and straightforward to implement, results in good agreement with the theory could be achieved.

A major drawback, aside from a missing local mesh refinement to reduce the number of grid points in the inlet and outlet regions, was the specific way diffusion was modelled as a passive scalar. More advanced models are available today (see e.g. [212, 213]) which avoid stability problems and allow to reach lower diffusion constants.

Concurrent Numerical Simulation of Flow and Clotting (Chap. 5.3)

In this chapter, a model to simulate milk clotting based on a residence time approach was applied. This very first numerical approach to the much more complex process of blood clotting¹³ allowed to produce results which were in good qualitative agreement with the experiments. Although a highly performance optimised sparse Lattice-Boltzmann code was used, simulating a few seconds of the clotting process required days on a NEC SX-8

¹²These studies were carried out by the author in the framework of a joint research with a Japanese automotive company, a presentation at the JSAE conference in Tokyo is planned for December 2007.

¹³The further progress towards modelling of blood clotting and its application to simulate thrombosis are not presented in this thesis, related publications to which the author contributed are [15, 173, 174].

vector computer. A significant drawback was again the limited stability of the diffusion model, which resulted in a far too high lower limit of the diffusion constant. As in the previous example, more advanced diffusion models could help to improve the situation.

The transient flow through an idealised stenosis (Chap. 5.3.4) possibly suffered from the insufficient approximation of the geometry by the voxel mesh. Applying improved wall boundary conditions for a better representation of the curvilinear boundary might have an influence on the flow patterns particularly in the vicinity of the stenosis, where the flow structures seem to be aligned with the mesh.

6.5.2 Things I did not do ...

Many fields of the current research within the very active Lattice-Boltzmann community have not been mentioned in this thesis: multi-phase flow [214, 215, 216, 217, 218, 219, 220, 221, 222], fluid-structure interaction [223, 224], particle transport [225, 226, 227, 228, 229, 230, 231, 232, 233] and shallow water and free surface simulation [234, 235], just to mention the most prominent. Also this work was restricted to implementations of the BGK relaxation operator, standard bounce-back wall boundary conditions, omitting local mesh refinement and turbulence modelling [132, 191, 236, 237, 192].

A detailed analysis of all possible extensions and improvements of these basic models would justify a thesis on its own, this restriction therefore seemed necessary to keep the scope of this work within reasonable limits. By partially referring to research results published years ago, when these models were emerging, this thesis provides a very conservative estimation of the applicability of the Lattice-Boltzmann method. Particularly with parallel sparse-matrix HPC implementations providing the MRT relaxation scheme in combination with improved wall boundary conditions and local mesh refinement, a new class of problems could be addressed.

Although a state-of-the-art Lattice-Boltzmann flow solver should offer a few of the above mentioned new functionalities, the majority of the currently used implementations are based on a simple bounce-back BGK scheme. The results presented in this work have relevance also for the planning of future studies, since they clearly demonstrate the power and some limitations of this basic, but very efficient approach. While the BGK scheme can, and should, be replaced by the more advanced TRT or MRT relaxation operators [31], or in the future even with the new cascaded model [238], codes with simple bounce-back wall boundary conditions not offering local mesh refinement will persist in the future, at least for multi-physics modelling.

As a final conclusion, the author would like to express his gratitude for the opportunity to contribute to two exciting research projects aiming for an improvement of human health – a very satisfying experience for a researcher active at the borderline of Fluid Dynamics and Computer Science.

Bibliography

- [1] K. Tanimura, K. Kozo, I. Kitahara, M. Matsumoto, and A. Akira. *The Encyclopedia* of Musical Instruments: *The Shakuhachi*. Tokyo Ongaku Sha, 1990.
- [2] http://www.navaching.com/shaku/bits.html Navaching. Shaku design various bits, 2003.
- [3] H. Kühnelt. Simulating the mechanism of sound generation in flutes using the lattice Boltzmann method. In *Proceedings of the Stockholm Musical Acoustics Conference, August 6-9, 2003 (SMAC03), Stockholm, Schweden*, pages 401–404. Royal Swedish Academy of Music, 2003.
- [4] H. Kühnelt. Simulating the sound generation in flue pipes with the lattice-Boltzmann method. In Proceedings of the International Symposium on Musical Acoustics, March 31st to April 3rd 2004 (ISMA2004), Nara, Japan, pages 251–254, 2004.
- [5] F. Durst, A. Melling, and J. H. Whitelaw. *Principles and Practice of Laser-Doppler Anemometry*. Academic Press, London, 1976.
- [6] F. Durst, A. Al-Salaymeh, P. Bradshaw, and J. Jovanovic. The development of a pulsed-wire probe for measuring flow velocity with a wide bandwidth. *Int. J. Heat and Fluid Flow*, 24(1), 2003.
- [7] O. Filippova and D. Hänel. Grid refinement for lattice-BGK models. J. Comput. *Phys.*, 147:219–228, 1998.
- [8] M. Krafczyk. *Gitter-Boltzmann-Methoden: Von der Theorie zur Anwendung (Habilitationsschrift, TU München).* München, 2001.
- [9] D. H. Rothman and S. Zaleski. Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics. Collection Aléa-Saclay: Monographs and Texts in Statistical Physics. Cambridge University Press, Cambridge, United Kingdom, 1997.
- [10] B. Chopard and M. Droz. Cellular Automata Modeling of Physical Systems. Collection Aléa-Saclay: Monographs and Texts in Statistical Physics. Cambridge University Press, Cambridge, United Kingdom, 1998.
- [11] D. A. Wolf-Gladrow. Lattice-Gas Automata and Lattice Boltzmann Models, volume 1725 of Lecture Notes in Mathematics. Springer, 2000.

- [12] J.-P. Rivet and J. P. Boon. *Lattice Gas Hydrodynamics*, volume 11 of *Cambridge Nonlinear Science Series*. Cambridge University Press, Cambridge, 2001.
- [13] S. Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Numerical Mathematics and Scientific Computation. Clarendon Press, Oxford, 2001.
- [14] F. Durst, R. Haas, and W. Interthal. The nature of flows through porous media. J. Non-Newtonian Fluid Mech., 22:169–189, 1987.
- [15] S. E. Harrison. *The Use of the Lattice Boltzmann Method in Thrombosis Modelling*. PhD thesis, Medical Physics and Clinical Engineering, University of Sheffield, 2007.
- [16] P. M. A. Sloot, J. A. Kaandorp, A. G. Hoekstra, and B. J. Overeinder. *Distributed Cellular Automata : Large Scale Simulation of Natural Phenomena*, chapter 1: Solutions to Parallel and Distributed Computing Problems: Lessons from Biological Sciences, pages 1–46. John Wiley and Sons, 2001.
- [17] P. M. A. Sloot and A. G. Hoekstra. *Handbook of Dynamic System Modeling*, chapter Modeling Dynamic Systems with Cellular Automata. CRC Computer & Information Science Series. Chapman & Hall, 2007.
- [18] S. Ulam. Random processes and transformations. *Proc. Int. Congr. Math.*, 2:264–275, 1952.
- [19] A. W. Burks, editor. *Essays on Cellular Automata*, chapter Von Neumann's self-reproducing automata, pages 3–64. University of Illinois Press, 1970.
- [20] M. Gardner. The fantastic combinations of John Conway's new solitaire game life. *Scientific American*, 220(4):120, 1970.
- [21] S. Wolfram. Theory and Applications of Cellular Automata including selected papers 1983-1986, volume 1 of Advanced series on complex systems. World Scientific, 1986.
- [22] S. Wolfram. *Cellular Automata and Complexity*. Addison-Wesley, Reading MA, 1994.
- [23] T. Toffoli and N. H. Margolus. *Cellular Automata Machines a New Environment for Modeling*. MIT Press, 1987.
- [24] J. Hardy, Y. Pomeau, and O. de Pazzis. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Phys. Rev. A*, 13:1949– 1960, 1976.
- [25] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier-Stokes Equation. *Phys. Rev. Lett.*, 56(14):1505–1508, 1986.
- [26] S. Wolfram. Cellular automaton fluid: basic theory. J. Stat. Phys., 45:471, 1986.
- [27] G. R. McNamara and G. Zanetti. Use of the Boltzmann equation to simulate lattice gas automata. *Phys. Rev. Lett.*, 61:2332–2335, 1988.

- [28] F. J. Higuera and J. Jiménez. Boltzmann approach to lattice gas simulations. *Europhys. Lett.*, 9(7):663–668, August 1989.
- [29] Y. H. Qian, D. d'Humières, and P. Lallemand. Lattice BGK models for Navier-Stokes equation. *Europhys. Lett.*, 17(6):479–484, 1992.
- [30] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo. Multiplerelaxation-time lattice Boltzmann models in 3D. ICASE Report No. 2002-20 NASA/CR-2002-211660, ICASE, Hampton, Virginia, 2002.
- [31] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo. Multiplerelaxation-time lattice Boltzmann models in three-dimensions. *Transactions of Royal Society of London A*, 360(1792):437–451, 2002.
- [32] D. P. Ziegler. Boundary conditions for lattice Boltzmann simulations. J. Stat. Phys., 71(5/6):1171–1177, 1993.
- [33] P. A. Skordos. Initial and boundary conditions for the lattice Boltzmann method. *Phys. Rev. E*, 48:4823–4842, 1993.
- [34] D. R. Noble, S. Chen, J. G. Georgiadis, and R. O. Buckius. A consistent hydrodynamic boundary condition for the lattice Boltzmann method. *Phys. Fluids*, 7(1):203– 209, January 1995.
- [35] D. R. Noble, J. G. Georgiadis, and R. O. Buckius. Direct assessment of lattice Boltzmann hydrodynamics and boundary conditions for recirculating flows. J. Stat. Phys., 81(1/2):17–33, 1995.
- [36] F. Nannelli and S. Succi. The lattice Boltzmann equation on irregular lattices. J. *Stat. Phys.*, 68(3/4):401–407, 1992.
- [37] X. He, L.-S. Luo, and M. Dembo. Some progress in lattice Boltzmann method. part I. nonuniform mesh grids. J. Comput. Phys., 129:357–363, 1996.
- [38] J. Bernsdorf and M. Schäfer. Practical aspects of the simulation of viscous flow using lattice Boltzmann automata. In R. Helmig, W. Jäger, W. Kinzelback, P. Knabner, and G. Wittum, editors, *Notes on numerical fluid mechanics: Modeling and comutation in environmental sciences*, volume 59, pages 61–71. Vieweg, 1997.
- [39] J. Bernsdorf, F. Durst, and M. Schäfer. Comparison of cellular automata and finite volume techniques for simulation of incompressible flows in complex geometries. *Int. J. Numer. Met. Fluids*, 29:251–264, 1999.
- [40] S. Succi, R. Benzi, and F. Higuera. The lattice Boltzmann equation: A new tool for computational fluid-dynamics. *Physica D*, 47:219–230, 1991.
- [41] J. A. Kaandorp, C. P. Lowe, D. Frenkel, and P. M. A. Sloot. Effect of nutrient diffusion and flow on coral morphology. *Phys. Rev. Lett.*, 77(11):2328–2331, 1996.
- [42] J. A. Kaandorp and P. M. A. Sloot. Morphological models of radiate accretive growth and the influence of hydrodynamics. J. Theor. Biol., 209:257–274, 2001.

- [43] G. A. Kohring. Effect of finite grain size on the simulation of fluid flow in porous media. J. Phys. II France, 1:87–90, 1991.
- [44] G. A. Kohring. The cellular automata approach to simulating fluid flows in porous media. *Physica A*, 186:97–108, 1992.
- [45] A. Koponen, D. Kandhai, E. Hellén, M. Alava, A. Hoekstra, M. Kataja, K. Niskanen, P. M. A. Sloot, and J. Timonen. Permeability of three-dimensional random fiber webs. *Phys. Rev. Lett.*, 80(4):716–719, 1998.
- [46] D. R. Noble, J. G. Georgiadis, and R. O. Buckius. Comparison of accuacy and performance for lattice Boltzmann and finite difference simulations of steady viscous flow. *Int. J. Numer. Met. Fluids*, 23:1–18, 1996.
- [47] T. Inamuro, M. Yoshino, and F. Ogino. A non-slip boundary condition for lattice Boltzmann simulations. *Phys. Fluids*, 7(12):2928–2930, 1997.
- [48] T. Inamuro, M. Yoshino, and F. Ogino. Accuracy of the lattice Boltzmann method for small Knudsen number with finite Reynolds number. *Phys. Fluids*, 9(11):3535– 3542, 1997.
- [49] X. He, Q. Zou, L.-S. Luo, and M. Dembo. Analytic solutions of simple flows and analysis of noneslip boundary conditions for the lattice Boltzmann BGK model. J. Stat. Phys., 87(1/2):115–136, 1997.
- [50] D. Kandhai, A. Koponen, A. G. Hoekstra, M. Kataja, J. Timonen, and P. M. A. Sloot. Implementation aspects of 3D lattice-BGK: Boundaries, accuracy, and a new fast relaxation method. *J. Comput. Phys.*, 150:482–501, 1999.
- [51] B. Ferréol and D. H. Rothman. Lattice-Boltzmann simulations of flow through Fontainebleau sandstone. *Transport in Porous Media*, 20(1-2):3–20, 1995.
- [52] C. Pan, L.-S. Luo, and C. T. Miller. An evaluation of lattice Boltzmann schemes for porous medium flow simulation. *Computers & Fluids*, 35:898–909, 2006.
- [53] M. Bouzidi, M. Firdaouss, and P. Lallemand. Momentum transfer of a Boltzmannlattice fluid with boundaries. *Phys. Fluids*, 13(11):3452–3459, 2001.
- [54] P. Lallemand and L.-S. Luo. Lattice Boltzmann method for moving boundaries. J. Comput. Phys., 184:406–421, 2003.
- [55] I. Ginzburg and D. d'Humières. Multireflection boundary conditions for lattice Boltzmann models. *Phys. Rev. E*, 68(066614):1–30, 2003.
- [56] J. Kropp. Analyse der inhomogenen Elementverteilung in Wattsedimenten. Master's thesis, Universität Oldenburg, 1992.
- [57] J. Bernsdorf, G. Brenner, T. Zeiser, P. Lammers, and F. Durst. Numerical analysis of the pressure drop in porous media flow using the lattice Boltzmann computational

technique. In N. Satofuka, editor, *Computational Fluid Dynamics 2000, Proceedings of the First International Conference on Computational Fluid Dynamics, Kyoto, Japan, 10-14 July 2000*, pages 493–498. Springer, 2001.

- [58] T. Zeiser, P. Lammers, E. Klemm, Y. W. Li, J. Bernsdorf, and G. Brenner. CFDcalculation of flow, dispersion and reaction in a catalyst filled tube by the lattice Boltzmann method. *Chem. Engng. Sci*, 56(4):1697–1704, 2001.
- [59] O. Filippova and D. Hänel. Boundary-fitting and local grid refinement for LBGK models. *Int. Journal of Modern Physics C*, 8:1271, 1998.
- [60] D. Kandhai, W. Soll, S. Chenc, A. Hoekstraa, and P. M. A. Sloot. Finitedifference lattice-BGK methods on nested grids. *Computer Physics Communications*, 129:100–109, 2000.
- [61] A. Dupuis and B. Chopard. Theory and applications of an alternative lattice Boltzmann grid refinement algorithm. *Phys. Rev. E*, 67:066707–1–7, 2003.
- [62] B. Crouse, E. Rank, M. Krafczyk, and J. Tölke. A LB-based approach for adaptive flow simulations. *Int. Journal of Modern Physics B*, 17:109–112, 2002.
- [63] B. Crouse. *Lattice-Boltzmann Strömungssimulationen auf Baumdaten-Strukturen*. PhD thesis, TU München, 2002.
- [64] R. Mei and W. Shyy. On the finite difference-base lattice Boltzmann method in curvilinear coordinates. *J. Comput. Phys.*, 143:426–448, 1998.
- [65] R. Mei, L.-S. Luo, and W. Shyy. An accurate curved boundary treatement in the lattice Boltzmann method. *J. Comput. Phys.*, 155:307–330, 1999.
- [66] X. He and L.-S. Luo. A priori derivation of the lattice Boltzmann equation. *Phys. Rev. E*, 55(6):R6333–R6336, June 1997.
- [67] X. He and L.-S. Luo. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. *Phys. Rev. E*, 56(6):6811–6817, December 1997.
- [68] P. Bhatnagar, E. P. Gross, and M. K. Krook. A model for collision processes in gases. I. small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94(3):511–525, May 1954.
- [69] J. J. Derksen and H. E. A. van den Akker. Large-eddy simulations on the flow driven by a Rushton turbine. *AIChE Journal*, 45:209–221, 1999.
- [70] J. J. Derksen. Assessment of large eddy simulations for agitated flows. *ChERD*, 79:824–830, 2001.
- [71] Z. Y. Lu, Y. Liao, D. Y. Qian, J. B. McLaughlin, J. J. Derksen, and K. Kontomaris. Large eddy simulations of a stirred tank using the lattice-Boltzmann method on a nonuniform grid. *J. Comput. Phys.*, 181:675–704, 2002.

- [72] J. J. Derksen. Lattice-Boltzmann based large-eddy simulations applied to industrial processes. In P. M. A. Sloot et al., editor, *Lecture Notes in Computer Science*, volume 2329, pages 713–722, 2002.
- [73] J. J. Derksen, K. Kontomaris, J. B. McLaughlin, and H.E.A. van den Akker. Largeeddy simulations of single-phase flow dynamics and mixing in an industrial crystallizer. *ChERD*, 85(A2):1–12, 2007.
- [74] G. Wellein, T. Zeiser, P. Lammers, and U. Küster. Towards optimal performance for lattice Boltzmann applications on terascale computers. In A. Deane, G. Brenner, and A. Ecer et al., editors, *Parallel Computational Fluid Dynamics: Theory and Applications, Proceedings of the 2005 International Conference on Parallel Computational Fluid Dynamics, May 24-27, College Park, MD, USA*, pages 31–40. Elsevier, 2006.
- [75] J. Bernsdorf, S. E. Harrison, S. M. Smith, P. V. Lawford, and D. R. Hose. A lattice-Boltzmann HPC application in medical physics. In M. Resch, Th. Bönisch, S. Tiyyagura, T. Furui, Y. Seo, and W. Bez, editors, *High Performance Computing on Vector Systems 2006*, Proceedings of the High Performance Computing Center Stuttgart, pages 121–132. Springer, 2006.
- [76] S. E. Harrison, J. Bernsdorf, D. R. Hose, and P. V. Lawford. A lattice Boltzmann framework for simulation of thrombogenesis. *Progress of CFD*, accepted for publication 2007.
- [77] J. Bernsdorf, S. E. Harrison, S. M. Smith, P. V. Lawford, and D. R. Hose. Applying the lattice Boltzmann technique to biofluids: A novel approach to simulate blood coagulation. *Computers and Mathematics with Applications*, in press 2007.
- [78] L. Axner, J. Bernsdorf, T. Zeiser, P. Lammers, J. Linxweiler, and A. G. Hoekstra. Performance evaluation of a parallel sparse lattice Boltzmann solver. *J. Comput. Phys.*, submitted 2007.
- [79] M. Schulz, M. Krafczyk, J. Tölke, and E. Rank. Parallelization strategies and efficiency of CFD computations in complex geometries using the lattice Boltzmann methods on high-performance computers. In M. Breuer, F. Durst, and C. Zenger, editors, *High Performance Scientific and Engineering Computing, Proceedings of the 3rd International Fortwihr Conference on HPSEC, Erlangen, March 12-14, 2001*, volume 21 of *Lecture Notes in Computational Science and Engineering*, pages 114–122. Springer, 2002.
- [80] G. Wellein, T. Zeiser, G. Hager, and S. Donath. On the single processor performance of simple lattice Boltzmann kernels. *Computers & Fluids*, 35:910–919, 2006.
- [81] L. Axner. High Performance Computational Hemodynamics with the Lattice Boltzmann Method. PhD thesis, Section Computational Science, University of Amsterdam, December 2007.
- [82] G. Karypis. METIS serial graph partitioning and fill-reducing matrix ordering, http://glaros.dtc.umn.edu/gkhome/metis/metis/overview, 11 1998.

- [83] J.-C. Desplat, I. Pagonabarraga, and P. Bladon. LUDWIG: A parallel lattice-Boltzmann code for complex fluids. *Computer Physics Communications*, 134:273– 290, 2001.
- [84] J. Wilke, T. Pohl, M Kowarschik, and U. Rüde. Cache performance optimizations for parallel lattice Boltzmann codes in 2D. *Parallel Processing*, pages 441–450, 2003.
- [85] MPI Forum. Mpi documents, http://www.mpi-forum.org/docs/docs.html.
- [86] L. Abrahamyan, J. Bernsdorf, T. Zeiser, P. Lammers, J. Linxweiler, A. G. Hoekstra, and P. M. A. Sloot. MPI parallelization and performance aspects of a sparse-data lattice boltzmann flow solver. Poster presented at the ICMMES 2006 - International Conference for Mesoscopic Methods in Engineering and Science, Hampton, USA, July 25-28, 2006, 2006.
- [87] J. Bernsdorf. anb Anb is Not Best, a Lattice-Boltzmann teaching code. http://www.bernsdorf.org/research/anb/, 1998.
- [88] T. Zeiser. priv. com., September 2007.
- [89] W. Li, X. Wei, and A. Kaufman. Implementing lattice Boltzmann computation on graphics hardware. *The Visual Computer*, 19(7-8):444–456, 2003.
- [90] N. S.-H. Chu and C.-L. Tai. Moxi: Real-time ink dispersion in absorbent paper. *ACM Transactions on Graphics (SIGGRAPH 2005 issue)*, 24(3):504 – 511, 2005.
- [91] J. Tölke. Implementation of a lattice Boltzmann kernel using the compute unified device architecture. *Computing and Visualization in Science*, submitted 2007.
- [92] J. Linxweiler, J. Tölke, and M. Krafczyk. Applying modern soft- and hardware technologies for computational steering approaches in computational fluid dynamics. in press 2007.
- [93] N. Margolus, T. Toffoli, and G. Vichniac. Cellular-automata supercomputers for fluid-dynamics modelling. *Phys. Rev. E*, 56(16):1694–1696, 1986.
- [94] N. Margolus. Lattice Gas Automata, chapter CAM-8: a computer architecture based on cellular automata, pages 167–188. Fields Institute Communications. American Mathematical Society, 1993.
- [95] Exa Corporation. Digital physics technology, principals & applications. Technical report, Exa Corporation, August 1994.
- [96] Business Wire. Exa and Sun microsystems form strategic aliancle. *Business Wire*, November 12. 1996.
- [97] J. Tölke. priv. com., Sept. 2007.

- [98] J. Bernsdorf, G. Brenner, S. Doi, H. Matsumoto, K. Nishizawa, T. Takei, and T. Zeiser. Visualisation of reactive flows in environmental sciences and chemical engineering using RVSLIB and lattice Boltzmann flow solver. In *Proceedings of the First International Symposium on Advanced Fluid Information, Sendai, Japan, October 4-5, 2001*, pages 292–395, 2001.
- [99] J. Bernsdorf, U. Jaekel, T. Zeiser, S. Doi, T. Takei, H. Matsumoto, and K. Nishizawa. Numerical simuration of reaction-diffusion and adsorption processes in porous media using lattice Boltzmann methods with concurrent visualisation. In K. Matsuno, A. Ecer, N. Satofuka, J .Periaux, and P. Fox, editors, *Parallel Computational Fluid Dynamics 2002, Proceedings of the Parallel CFD 2002 Conference, Kansai Science City, Japan, May 20-22, 2002*, pages 233–240. Elsevier, 2003.
- [100] J. Bernsdorf, U. Jaekel, T. Zeiser, T. Takei, H. Matsumoto, and K. Nishizawa. Lattice Boltzmann simulation and visualisation of adsorption processes in complex geometries. In P. M. A. Sloot, D. Abramson, A. V. Bogdanov, J. Dongarra, A. Y. Zomaya, and Y. E. Gorbachev, editors, *Computational Science - ICCS 2003, International Conference, Melbourne, Australia and St. Petersburg, Russia, June 2-4, 2003. Proceedings, Part I*, volume 2657 of *Lecture Notes in Computer Science*, pages 1054– 1061. Springer, 2003.
- [101] T. Takei, J. Bernsdorf, N. Masuda, and H. Takahara. Lattice Boltzmann simulation and its concurrent visualization on the SX-6 supercomputer. In *HPCASIA '04: Proceedings of the High Performance Computing and Grid in Asia Pacific Region, Seventh International Conference on (HPCAsia'04)*, pages 212–219, Washington, DC, USA, 2004. IEEE Computer Society.
- [102] R. G. Belleman and R. Shulakov. High performance distributed simulation for interactive simulated vascular reconstruction. In P. M. A. Sloot, C. J. K. Tan, J. J. Dongarra, and A. G. Hoekstra, editors, *Computational Science - ICCS 2002, Proceedings Part III*, volume 2331 of *Lecture Notes in Computer Science*, pages 265– 274. Springer, 2002.
- [103] R. G. Belleman and P. M. A. Sloot. The design of dynamic exploration environments for computational steering simulations. In Marian Bubak et al., editor, SGI Users' Conference, pages 57–74, Krakow, 2000.
- [104] A. Borrmann, P. Wenisch, C. van Treeck, and E. Rank. Collaborative computational steering: Principles and application in HVAC layout. *Integrated Computer-Aided Engineering*, 13(4):361–376, 2006.
- [105] B. Crouse, M. Krafczyk, S. Kühner, E. Rank, and C. van Treeck. Indoor air flow analysis based on lattice Boltzmann methods. *Energy and Buildings*, 34:941–949, 2002.
- [106] T. Takei, H. Matsumoto, K. Muramatsu, and S. Doi. Parallel vector performance of concurrent visualization system RVSLIB on SX-4. *Proc. 3rd Pacific Symposium on Flow Visualization and Image Processing*, 2001.

- [107] S. Geller, M. Krafczyk, J. Tölke, S. Turek, and J. Hron. Benchmark computations based on lattice-Boltzmann, finite element and finite volume methods for laminar flows. *Computers & Fluids*, 35:888–897, 2006.
- [108] B. D. Kandhai, D. J.-E. Vidal, A. G. Hoekstra, H. C. J. Hoefsloot, P. Iedema, and P. M. A. Sloot. A comparison between lattice-Boltzmann and finite element simulations of fluid flow in static mixer reactors. *Int. Journal of Modern Physics C*, 9:1123–1128, 1998.
- [109] J. Bernsdorf, T. Zeiser, G. Brenner, and F. Durst. Simulation of a 2D channel flow around a square obstacle with lattice-Boltzmann (BGK)-automata. *Int. Journal of Modern Physics C*, 9(8):1129–1141, December 1998.
- [110] H. Yu, S. S. Girimaji, and L.-S. Luo. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. J. Comput. Phys., 206:599–616, 2005.
- [111] D. O. Martinez, W. H. Matthaeus, S. Chen, and D. C. Montgomery. Comparison of spectral method and lattice Boltzmann simulations of two-dimensional hydrodynamics. *Phys. Fluids*, 6(3):1285–1298, March 1994.
- [112] S. Chen, Z. Wang, X. Shan, and G. D. Doolen. Lattice Boltzmann computational fluid dynamics in three dimensions. *J. Stat. Phys.*, 68:379–400, 1992.
- [113] T. Poinsot and F. Nicoud. Etude de faisabilité d'une simulation numerique directe d'un ecoulement avec forte injection paretale. Technical report, CERFACS technical report, 1994.
- [114] G. Brenner, T. Schönfeld, and N. Mangiavacchi. A multiphysics approach to the direct numerical simulation of turbulent flows on parallel computers. In *HPCN Europe 1996: Proceedings of the International Conference and Exhibition on High-Performance Computing and Networking*, volume 1067, pages 607 – 612, 1996.
- [115] J. O. Hinze. Turbulence. McGraw-Hill, 1975.
- [116] T. Zeiser. Numerische Simulation der instationären Umströmung eines quadratischen Zylinders mit einem Lattice-Boltzmann Verfahren. Studienarbeit, Lehrstuhl für Strömungsmechanik, Universität Erlangen-Nürnberg, Erlangen, Germany, 1998.
- [117] M. Breuer, J. Bernsdorf, T. Zeiser, and F. Durst. Accurate computations of the laminar flow past a square cylinder bases on two different methods: Lattice-Boltzmann and Finite-Volume. *Int. J. Heat and Fluid Flow*, 21(2):186–196, 2000.
- [118] C. H. K. Williamson. Vortex dynamics in the cylinder wake. Annu. Rev. Fluid Mech., 28:477–539, 1996.
- [119] M. M. Zdravkovich. *Flow Around Circular Cylinders, Vol. 1: Fundamentals*. Oxford University Press, New York, 1997.

- [120] R. Franke, W. Rodi, and B. Schönung. Numerical calculation of laminar vortex shedding flow past cylinders. *J. Wind Eng. Ind. Aerody.*, 35:237–257, 1990.
- [121] K. M. Kelkar and S. V. Patankar. Numerical prediction of vortex shedding behind a square cylinder. *Int. J. Numer. Met. Fluids*, 14:327–341, 1992.
- [122] R. W. Davis and E. F. Moore. A numerical study of vortex shedding from rectangles. *J. Fluid Mech.*, 116:475–506, 1982.
- [123] R. W. Davis, E. F. Moore, and L. P. Purtell. A numerical-experimental study of confined flow around rectangular cylinders. *Phys. Fluids*, 27(1):46–59, 1984.
- [124] A. Okajima. Strouhal numbers of rectangular cylinders. J. Fluid Mech., 123:379–398, 1982.
- [125] A. Okajima. Numerical simulation of flow around rectangular cylinders. J. Wind Eng. Ind. Aerody., 33:171–180, 1990.
- [126] A. Mukhopadhyay, G. Biswas, and T. Sundararajan. Numerical investigation of confined wakes behind a square cylinder in a channel. *Int. J. Numer. Met. Fluids*, 14:1473–1484, 1992.
- [127] H. Suzuki, Y. Inoue, T. Nishimura, K. Fukutani, and K. Suzuki. Unsteady flow in a channel obstructed by a square rod (crisscross motion of vortex). *Int. J. Heat and Fluid Flow*, 14(1):2–9, 1993.
- [128] R. Franke. Numerische Berechnung der instationären Wirbelablösung hinter zylindrischen Körpern. PhD thesis, University of Karlsruhe, 1991.
- [129] G. Li and J. A. C. Humphrey. Numerical modelling of confined flow past a cylinder of square cross-section at various orientations. *Int. J. Numer. Met. Fluids*, 20:1215– 1236, 1995.
- [130] F. H. Shair, A. S. Grove, E. E. Petersen, and A. Acrivos. The effect of confining walls on the stability of the steady wake behind a circular cylinder. *J. Fluid Mech.*, 17:546–550, 1963.
- [131] J. G. M. Eggels and J. A. Somers. Numerical simulation of free convective flow using lattice-Boltzmann scheme. *Int. J. Heat and Fluid Flow*, 16(5):357–364, 1995.
- [132] J. G. M. Eggels. Direct and large-eddy simulation of turbulent fluid flow using the lattice-Boltzmann scheme. *Int. J. Heat and Fluid Flow*, 17:307–323, 1996.
- [133] M. Breuer and W. Rodi. Large–eddy simulation of turbulent flow through a straight square duct and a 180° bend. In P. R. Voke, L. Kleiser, and J. P. Chollet, editors, *Fluid Mechanics and its Applications: Direct and Large–Eddy Simulation I, Selected Papers from the First ERCOFTAC Workshop on Direct and Large–Eddy Simulation, Guildford, Surrey, UK, March 27–30, 1994*, volume 26, pages 273–285. Kluwer Academic Publishers, Dordrecht, 1994.

- [134] M. Breuer, D. Lakehal, and W. Rodi. Flow around a surface mounted cubical obstacle: Comparison of LES and RANS-results. In M. Deville, S. Gavrilakis, and I. L. Rhyming, editors, *Notes on Numerical Fluid Mechanics: Computation of 3D Complex Flows*, volume 53, pages 22–30. Vieweg Verlag, Braunschweig, 1996.
- [135] M. Breuer and M. Pourquié. First experiences with LES of flows past bluff bodies. In W. Rodi and G. Bergeles, editors, Proc. of the 3rd International Symposium of Engineering Turbulence Modelling and Measurements, Heraklion, Crete, Greece, May 27–29, 1996, Engineering Turbulence Modelling and Experiments 3, pages 177–186. Elsevier Science B.V., Amsterdam, 1996.
- [136] M. Breuer and W. Rodi. Large–eddy simulation of complex turbulent flows of practical interest. In E. H. Hirschel, editor, *Notes on Numerical Fluid Mechanics: Flow Simulation with High–Performance Computers II*, volume 52, pages 258–274. Vieweg Verlag, Braunschweig, 1996.
- [137] M. Breuer. Numerical and modeling influences on large eddy simulations for the flow past a circular cylinder. In E. H. Hirschel, editor, *Proc. of the 11th Symposium on Turbulent Shear Flows, Grenoble, France, Sept. 8–11, 1997*, volume 3, pages 26:7–26:12, 1997.
- [138] M. Breuer. Numerical and modeling influences on large eddy simulations for the flow past a circular cylinder. *Int. J. Heat and Fluid Flow*, 19/5:512–521, 1997.
- [139] M. Breuer. Large eddy simulation of the sub-critical flow past a circular cylinder: Numerical and modeling aspects. *Int. J. Numer. Met. Fluids*, 28:1281–1302, 1998.
- [140] M. Breuer. Large eddy simulation of high reynolds number circular cylinder flow. In A. Tezel S. Biringen, H. Örs and J. H. Ferziger, editors, *Lecture Notes in Physics, Proc. of the Workshop on Industrial and Environmental Applications of Direct and Large–Eddy Simulation, Aug. 5–7, 1998, Boğaziçi University, Istanbul, Turkey*, volume 529, pages 176–189. Springer Verlag, 1999.
- [141] C. M. Rhie and W. L. Chow. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA Journal*, 21:1225–1532, 1983.
- [142] T. F. Miller and F. W. Schmidt. Use of pressure–weighted interpolation method for the solution of the incompressible Navier–Stokes equations on a non–staggered grid system. *Num. Heat Transfer*, 14:213–233, 1988.
- [143] M. H. Kobayashi and J. C. F. Pereira. Numerical comparison of momentum interpolation methods and pressure-velocity algorithms using non-staggered grids. *Commun. Appl. Numerical Methods*, 7:173–186, 1991.
- [144] R. S. Maier and R. S. Bernard. Accuracy of the lattice-Boltzmann method. *Int. Journal of Modern Physics C*, 8(4):747–752, 1997.
- [145] J. Bernsdorf, O. Günnewig, W. Hamm, and M. Münker. Strömungsberechnung in porösen Medien. GIT Labor-Fachzeitschrift, 4:387–390, 1999.

- [146] J. Bernsdorf, F. Delhopital, G. Brenner, and F. Durst. Prediction of pressure losses in porous media using the lattice Boltzmann method. In Keil, Mackens, Voß, and Werther, editors, *Scientific Computing in Chemical Engineering II, Computational Fluid Dynamics, Reaction Engineering, and Molecular Properties*, volume 1, pages 336–343. Springer, 1999.
- [147] J. Bernsdorf, G. Brenner, and F. Durst. Numerical analysis of the pressure drop in porous media flow with lattice Boltzmann (BGK) automata. *Computer Physics Communications*, pages 247–255, 2000.
- [148] D. S. Clague, B. D. Kandhai, R. Zhang, and P. M. A. Sloot. Hydraulic permeability of (un)bounded fibrous media using the lattice Boltzmann method. *Phys. Rev. E*, 61(1):616–625, 2000.
- [149] W. v. Engelhardt. Der Porenraum der Sedimente. Springer, Berlin, 1960.
- [150] J. Bear. *Dynamics of Fluids in Porous Media*. American Elseviewer Publishing Company, Inc., New York, London, Amsterdam, 1972.
- [151] A. E. Scheidegger. *The Physics of Flow through Porous Media*. Univ. of Toronto Press, Toronto, Canada, 1974.
- [152] M. Sahimi. *Flow and Transport in Porous Media and Fractured Rock.* VCH Verlagsgesellschaft, Weinheim, New York, Basel, Cambridge, Tokyo, 1995.
- [153] R. B. Bird, W. E. Stewared, and E. N. Lightfoot. *Transport Phenomena*, chapter 6, pages 180–207. John Wiley & Son, 1960.
- [154] A. Koponen, M. Kataja, and J. Timonen. Tortuous flow in porous media. *Phys. Rev.* E, 54(1):406–410, 1996.
- [155] B. Chopard and A. Masselot. Cellular automata and lattice Boltzmann methods: a new approach to computational fluid dynamics and particle transport. *Future Generation Computer Systems*, 16:249–257, 1999.
- [156] B. Chopard, A. Masselot, and A. Dupuis. A lattice gas model for erosion and particles transport in a fluid. *Computer Physics Communications*, 129:167–176, 2000.
- [157] A. Dupuis and B. Chopard. Lattice gas modeling of scour formation under submarine pipelines. J. Comput. Phys., 178:161174, 2002.
- [158] T. Zeiser, G. Brenner J. Bernsdorf, P. Lammers, and F. Durst. Performance aspects of lattice Boltzmann methods for applications in chemical engineering. In C. B. Jenssen et al., editor, *Parallel Computational Fluid Dynamics 2000, Trends and Applications, Proceedings of the Parallel CFD 2000 Conference, May 22-25, Trondheim, Norway*, pages 407–414. Elsevier, 2001.
- [159] T. Zeiser, H.-J. Freund, J. Bernsdorf, P. Lammers, G. Brenner, and F. Durst. Detailed simulation of transport processes in reacting multi-species flows through complex geometries by means of the lattice Boltzmann method. In E. Krause and W. Jäger,

editors, *High Performance Computing in Science and Engineering '01, Transactions of the High Performance Computing Center Stuttgart (HLRS) 2001*, pages 442–452. Springer, 2002.

- [160] T. Zeiser, M. Steven, H.-J. Freund, P. Lammers, G. Brenner, F. Durst, and J. Bernsdorf. Analysis of the flow field and pressure drop in fixed bed reactors with the help of lattice Boltzmann simulations. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 360(1792):507–520, 2002.
- [161] T. Zeiser, H.-J. Freund, J. Bernsdorf, Y.-W. Li, P. Lammers, G. Brenner, and F. Durst. CFD calculations of flow, dispersion and chemical reactions in fixed-bed tubular reactors using the lattice Boltzmann method. In M. Breuer, F. Durst, and Ch. Zenger, editors, *High Performance Scientific and Engineering Computing: Proceedings of the 3rd International FORTWIHR Conference on HPSEC*, volume 21 of *Lecture Notes in Computational Science and Engineering*, pages 53–62. Springer, 2002.
- [162] E. G. Flekkøy. Lattice Bhatnagar-Gross-Krook models for miscible fluids. *Phys. Rev. E*, 47(6):4247–4257, 1993.
- [163] U. Jaekel and H. Vereecken. Transport of solutes undergoing a Freundlich type nonlinear and non-equilibrium adsorption process. *Phys. Rev. E*, 65:041402(1–9), 2002.
- [164] A. M. Carmo, L. S. Hundal, and M. L. Thompson. Sorption of hydrophobic organic compounds by soil materials: Application of unit equivalent Freundlich sorption coefficients. *Environ. Sci. Technol.*, 34:4363–4369, 2000.
- [165] B. Xing, J. Pignatello, and B. Gigliotti. Competitive sorption between atrazine and other organic compounds in soils and model sorbents. *Environ. Sci. Technol.*, 30(2432-2440), 1996.
- [166] H. Vereecken, U. Jaekel, O. Esser, and O. Nitzsche. Solute analysis of bromide, uranin and LiCl using breakthrough curves from aquifer sediment. *Journal of Contaminant Hydrology*, 39:7–34, 1999.
- [167] R. Grundy, C. van Duijn, and C. Dawson. Asymptotic profiles with finite mass in one-dimensional contaminant transport through porous media: The fast reaction case. Q. J. Mech. Appl. Math., 47(1):69–106, 1994.
- [168] U. Jaekel, A. Georgescu, and H. Vereecken. Asymptotic analysis of nonlinear equilibrium transport in porous media. *Water Resour. Res.*, 36:3093–3098, 1996.
- [169] J. Bernsdorf, S. E. Harrison, D. R. Hose, P. V. Lawford, and S. M. Smith. Concurrent numerical simulation of flow and blood clotting using the lattice Boltzmann technique. In J. Ma and L. T. Yang, editors, *Proceedings of the 11th International Conference on Parallel and Distributed Systems: ICPADS 2005; Fukuoka, Japan, July 20-22, 2005*, volume II, pages 336–340, 2005.
- [170] J. Bernsdorf, S. E. Harrison, D. R. Hose, P. V. Lawford, and S. M. Smith. Numerical simulation of clotting processes: A lattice Boltzmann application in medical physics. *Mathematics and Computers in Simulation*, 72(2-6):89–92, 2006.

- [171] J. Bernsdorf, S. E. Harrison, D. R. Hose, P. V. Lawford, and S. M. Smith. Concurrent numerical simulation of flow and blood clotting using the lattice Boltzmann technique. *Int. J. Bioinformatics Research and Applications*, 2(4):371–380, 2006.
- [172] S. M. Smith. Development of a technique for the prediction of clot localisation in vitro. PhD thesis, Medical Physics and Clinical Engineering, University of Sheffield, Sheffield, 2005.
- [173] S. E. Harrison, S. M. Smith, J. Bernsdorf, D. R. Hose, and P. V. Lawford. Application and validation of the lattice Boltzmann method for modelling flow-related clotting. *Journal of Biomechanics*, in press 2007.
- [174] S. E. Harrison, J. Bernsdorf, D. R. Hose, and P. V. Lawford. Development of a lattice Boltzmann framework for numerical simulation of thrombosis. *Int. Journal of Modern Physics C*, 18(4):483–491, 2007.
- [175] World Health Organisation. Cardiovascular disease (2005). http://www.who.int/cardiovascular_diseases/resources/atlas/en, 2005.
- [176] L. Badimon and J. J. Badimon. Mechanisms of arterial thrombosis in non parallel streamlines: platelet thrombi grow on the apex of stenotic severely injured vessel wall. J. Clin. Invest., 84:1134–1144, 1989.
- [177] C. P. Markou, S. R. Hanson, J. M. Siegel, and D. N. Ku. The role of high wall shear rate on thrombus formation in stenoses. *Adv. Bioeng*, 26:555–558, 1993.
- [178] A. Tippe and H. Muller-Mohnssen. Shear dependence of the fibrin coagulation kinetics in vitro. *Thromb. Res*, 72:379–388, 1993.
- [179] K. Affeld and J. Gadischke. Shear rate and thrombin transport. *3rd International Symposium Reihe 17: Biotechnik*, 107:35–40, 1994.
- [180] K. Affeld, A. J. Reininger, J. Gadischke, K. Grunert, S. Schmidt, and F. Thiele. Fluid mechanics of the stagnation point flow chamber and its platelet deposition. *Artif. Organs*, 19(7):597–602, 1995.
- [181] V. I. Zarnitsina, A. V. Pokhilko, and F. I. Ataullakhanov. A mathematical model for the spatio-temporal dynamics of intrinsic pathway of blood coagulation. i. the model description. *Thromb. Res.*, 84(4):225–236, 1996.
- [182] V. I. Zarnitsina, A. V. Pokhilko, and F. I. Ataullakhanov. A mathematical model for the spatio-temporal dynamics of intrinsic pathway of blood coagulation. ii. results phenomenological model. *Thromb. Res.*, 84(5):333–344, 1996.
- [183] E. N. Sorensen, G. W. Burgreen, W. R. Wagner, and J. F. Antaki. Computational simulation of platelet deposition and activation: I. model development and properties. *Ann. Biomed. Eng.*, 27:436–448, 1999.
- [184] E. N. Sorensen, G. W. Burgreen, W. R. Wagner, and J. F. Antaki. Computational simulation of platelet deposition and activation: II. results for Poiseuille flow over collagen. *Ann. Biomed. Eng.*, 27:449–458, 1999.
- [185] F. I. Ataullakhanov, Y. V. Krasotkina, V. I. Sarbash, R. I. Volkova, E. I. Sinauridse, and A. Y. Kondratovich. Spatial-temporal dynamics of blood coagulation and pattern formation. an experiment study. *Int. J. Bifurcation and Chaos*, 12(9):1969–1984, 2002.
- [186] F. I. Ataullakhanov, V. I. Zarnitsina, A. V. Pokhilko, A. I. Lobanov, and O. L. Morozova. Spatio-temporal dynamics of blood coagulation and pattern formation. a theoretical approach. *Int. J. Bifurcation and Chaos*, 12:1985–2002, 2002.
- [187] R. Ouared and B. Chopard. Lattice Boltzmann simulations of blood flow non-Newtonian rheology and clotting processes. J. Stat. Phys., 121(1-2):209–221, 2005.
- [188] A. G. Hoekstra, B. Chopard, P. Lawford, D. R. Hose, M. Krafczyk, and J. Bernsdorf. Introducing complex automata for modelling multi-scale complex systems. In Proceedings of the European Conference on Complex Systems ECCS 2006, European Complex Systems Society, Oxford, UK, September 2006, 2006.
- [189] A. G. Hoekstra, E. Lorenz, J.-L. Falcone, and B. Chopard. Towards a complex automata framework for multi-scale modeling: Formalism and the scale separation map. In Y. Shi et al., editor, *ICCS 2007, Part I, LNCS 4487*, pages 922–930, Berlin, Heidelberg, 2007. Springer.
- [190] J. Tölke, M. Krafczyk, and E. Rank. A multigrid solver for the discrete Boltzmann equation, *J. Stat. Phys.*, 107(1/2):573–591, 2002.
- [191] S. Hou, J. D. Sterling, S. Chen, and G. D. Doolen. A lattice Boltzmann subgrid model for high Reynolds number flow. *Fields Institute Communications*, 6:151–166, 1996.
- [192] H. Chen, S. Succi, and S. Orszag. Analysis of sub-grid fluid turbulence using the Boltzmann Bhatnagar-Gross-Krook kinetic equation. *Phys. Rev. E*, 59(3), 1999.
- [193] O. Filippova, F. Mazzocco, S. Succi, C. Arrighetti, and G. Bella. Multiscale lattice Boltzmann schemes with turbulence modeling. J. Comput. Phys., 170(2):812–829, 2001.
- [194] EXA Corporation. http://www.exa.com.
- [195] MAGMA Giessereitechnologie GmbH. MAGMASOFT, www.magmasoft.de, status of 2007.
- [196] J. Latt, V. Heuveline, and M. Krause. openLB, www.openlb.org, status of 2007.
- [197] J. Bernsdorf, T. Zeiser, P. Lammers, G. Brenner, and F. Durst. Perspectives of the lattice Boltzmann method for industrial applications. In C.B. Jenssen et al., editor, *Parallel Computational Fluid Dynamics 2000, Trends and Applications, Proceedings of the Parallel CFD 2000 Conference, May 22-25, Trondheim, Norway*, pages 367–373. Elsevier, 2001.

- [198] D. Wang and J. Bernsdorf. Lattice Boltzmann simulations of steady shear thinning blood flow with a Carreau-Yasuda model. *Proceedings of the Parallel CFD 2007 Conference, Antalya, Turkey, May 16-20, 2007*, submitted 2007.
- [199] J. Bernsdorf and D. Wang. Non-Newtonian blood flow simulation in cerebral aneurysms. *Proceedings of the ICMMES 2007 Conference, Munich, Germany, July 16-20, 2007*, submitted 2007.
- [200] J. Bernsdorf and D. Wang. Blood flow simulation in cerebral aneurysm: A lattice Boltzmann application in medical physics. In *Proceedings of the Parallel CFD 2007 Conference, Antalya, Turkey, May 16-20, 2007*, submitted 2007.
- [201] The @neurIST Integrated Project. Project homepage: http://www.aneurist.org.
- [202] A. M. Artoli. *Mesoscopic Computational Haemodynamics*. PhD thesis, University of Amsterdam, 2003.
- [203] K. N. Beronov and F. Durst. Numerische Simulation pulsierender Strömungen in Blutgefässen des Gehirns mit Aneurysmen mittels Lattice-Boltzmann-Verfahren. Zeitschrift fuer Medizinische Physik, 15:257–264, 2005.
- [204] F. J. H. Gijsen, F. N. Van de Vosse, and J. D. Janssen. The influence of the non-Newtonian properties of blood on the flow in large arteries: steaday flow in a carotid bifurcation model. *J.Biomechanics*, 32:601–608, 1999.
- [205] S. Moore, T. David, J. G. Chase, J. Arnold, and J. Fink. 3D models of blood flow in the cerebral vasculature. *J.Biomechanics*, 39:1454–1463, 2006.
- [206] A. M. Artoli and A. Sequeir. Mesoscopic simulations of unsteady shear-thinning flows. In *Computational Science - ICCS 2006*, volume 3992/2006 of *Lecture Notes* in *Computer Science*, pages 78–85. Springer, 2006.
- [207] S. Shibeshi and W. E. Collins. The rheology of blood flow in a branched arterial system. *Appl. Rheol.*, 15(6):398–405, 2005.
- [208] T.-M. Liou and S.-N. Liou. A review on in vitro studies of hemodynamic characteristics in terminal and lateral aneurysm models. *Proc. Natl. Sci. Counc. ROC(B)*, 23(4):133–148, 1999.
- [209] S. Tateshima, Y. Maruyama, J. P. Villablanca, T. Morino, K. Nomura, K. Tanishita, and F. Vinuela. In vitro measurement of fluid-induced wall shear stress in unruptured cerebral aneurysms harbouring blebs. *Stroke*, 34:187–192, 2003.
- [210] The IST Project COAST. Project homepage: http://www.complex-automata.org.
- [211] R. Mai, L.-S. Luo, P. Lallemand, and D. d'Humières. Consistent initial conditions for lattice Boltzmann simulations. *Computers & Fluids*, 35:855–862, 2006.
- [212] I. Ginzburg. Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation. *Advances in Water Resources*, 28:1171–1195, 2005.

- [213] I. Ginzburg. Generic boundary conditions for lattice Boltzmann models and their application to advection and anisotropic dispersion equations. *Advances in Water Resources*, 28:1196–1216, 2005.
- [214] X. Shan and H. Chen. Lattice Boltzmann model for simulationg flows with multiple phases and components. *Phys. Rev. E*, 47(3):1815–1819, 1993.
- [215] D. Grunau, S. Chen, and K. Eggert. A lattice Boltzmann model for multiphase fluid flows. *Phys. Fluids A*, 5(10):2557–2562, 1994.
- [216] S. Chen, G. D. Doolen, and K. G. Eggert. Lattice-Boltzmann fluid dynamics a versatile tool for multiphase and other complicated flows. *Los Alamos Science*, 22:99–111, 1994.
- [217] J. F. Olson and D. H. Rothman. Three-dimensional immiscible lattice gas: Application to sheared phase separation. J. Stat. Phys., 81(1/2):199–222, 1995.
- [218] C. Appert, J. F. Olson, D.H. Rothman, and S. Zaleski. Phase separation in a threedimensional, two-phase hydrodynamic lattice gas. J. Stat. Phys., 81(1/2), 1995.
- [219] M.-L. Tan, Y. H. Qian, I. Goldhirsch, and S. A. Orszag. Lattice BGK approach to simulating granular flows. J. Stat. Phys., 81(1/2):87–103, 1995.
- [220] S. Hou, X. Shan, Q. Zou, G. D. Doolen, and W. E. Soll. Evaluation of two lattice Boltzmann models for multiphase flows. *J. Comput. Phys.*, 138:695–713, 1997.
- [221] M. Krafczyk, M. Schulz, and E. Rank. Lattice-gas simulations of two-phase flow in porous media. *Commun. numer. methods eng.*, 14(8):709–717, 1998.
- [222] T. Inamuro, T. Ogata, S. Tajima, and N. Konishi. A lattice Boltzmann method for incompressible two-phase flows with large density differences. J. Comput. Phys., 198(2):628–644, 2004.
- [223] M. Krafczyk, J. Tölke, E. Rank, and M. Schulz. Two-dimensional simulation of fluid-structure interaction using lattice-Boltzmann methods. *Computers and Structures*, 79:2031–2037, 2001.
- [224] S. Geller, J. Tölke, M. Krafczyk, D. Scholz, A. Düster, and E. Rank. Simulation of bidirectional fluid-structure interaction based on explicit coupling approaches of lattice Boltzmann and P-FEM solvers. In E. Onate M. Papadrakakis and B. Schrefler, editors, *Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2005*, Barcelona, 2005.
- [225] A. J. C. Ladd. Numerical simulations of particulate suspensions via a discrete Boltzmann equation. part 1. theoretical foundation. *J. Fluid Mech.*, 271:285–309, 1994.
- [226] A. J. C. Ladd. Numerical simulations of particulate suspensions via a discrete Boltzmann equation. part 2. numerical results. J. Fluid Mech., 271:311–339, 1994.
- [227] O. Behrend. Solid-fluid boundaries in particle suspension simulation vio the lattice Boltzmann method. *Phys. Rev. E*, 52(1):1164–1175, 1995.

- [228] C. K. Aidum and Y. Lu. Lattice Boltzmann simulation of solid particles suspended in fluid. J. Comput. Phys., 81(1/2):49–61, 1995.
- [229] C. K. Aidun, Y. Lu, and E.-J. Ding. Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation. *J. Fluid Mech.*, 373:287–311, 1998.
- [230] A. Masselot and B. Chopard. A lattice Boltzmann model for particle transport and deposition. *Europhys. Lett.*, 42(3):259–264, 1998.
- [231] D. Qi. Lattice-Boltzmann simulations of particles in non-zero-Reynolds-number flow. *J. Fluid Mech.*, 385:41–62, 1999.
- [232] J. Lee and J. Koplik. Microscopic motion of particles flowing through a porous medium. *Phys. Fluids*, 11(1):76–87, 1999.
- [233] A. J. C. Ladd. Lattice-Boltzmann simulations of hydrodynamically interacting particles. In H. van Beijeren and J. Karkheck, editors, *Dynamics: Models and Kinetik Methods for Non-equilibrium Many Body Systems*, pages 17–30, Kluwer Academic Publishers-Dotrecht, 2000.
- [234] N. Thürey, U. Rüde, and M. Stamminger. Animation of open water phenomena with coupled shallow water and free surface simulation. In M. P. Cani and J. O'Brien, editors, *Proceedings of the 2006 Eurographics/ACM SIGGRAPH Symposium on Computer Animation (ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Wien, 02. - 04.09. 2006)*, pages 157–166, 2006.
- [235] J. B. Frandsen. Free surface water wave 1-D LBGK predictions. *International Journal of Computational Fluid Dynamics*, 20(6):427–437, 2006.
- [236] C. M. Teixeira. Incorporating turbulence models into the lattice-Boltzmann method. *Int. Journal of Modern Physics C*, 9(8):1159–1175, 1998.
- [237] M. M. Pervaiz and C. M. Teixeira. Two equation turbulence modeling with the lattice Boltzmann method. Proceedings of the 2nd International Symosium on Computational Technologies for Fluid/Thermal/Chemical Systems with Industrial Applications, ASME PVP Division Conference, August 1-5 1999, Bosten, MA, 1999.
- [238] M. Geier, A. Greiner, and J. G. Korvink. Cascaded digital lattice Boltzmann automata for high Reynolds number flow. *Phys. Rev. E*, 73:066705, 2006.

Samenvatting

Dit proefschrift bevat in retrospect tien jaar onderzoek op het gebied van de vloeistofdynamica, chemische techniek en medische fysica. Door een grote verscheidenscheid aan voorbeelden uit het onderzoek van de auteur te bespreken, uitgevoerd binnen verschillende samenwerkingsverbanden, wordt de nadruk in dit proefschrift gelegd op: ten eerste, een bijdrage aan het onderzoek in de vloeistofmechanica en multi-fysica modellen, waarbij de Rooster Boltzmannmethode werd gebruikt als een krachtig gereedschap voor het uitvoeren van numerieke simulaties; en ten tweede, een demonstratie van het gebruik van de rooster Boltzmannmethode als een competatieve vloeistofsimulatiemethode, zowel in termen van rekenprestaties als in termen van gebruik in realistische stromingsproblemen.

Voor die toepassingen waar de traditionele CFD (Navier-Stokes) aanpak problemen ondervindt ten gevolge van complexe of variërende randen, of ten gevolgde van de noodzaak om de simulaties uit te breiden voor multi-fysica problemen, worden de voordelen van de rooster Boltzmannmethode aangetoond.

De keuze van de methode en de details van de implementatie werden in dit onderzoek bepaald door de specifieke toepassing. Bepaalde eigenschappen en hardware specifieke optimalisaties werden slechts dan geïmplementeerd wanneer deze nodig waren om een probleem op te lossen, en niet omdat ze mogelijk waren.

De rooster Boltzmannmethode wordt in hoofdstuk twee geïntroduceerd, inclusief een korte historie van de ontwikkeling van de methode en, om didactische redenen, een uitleg van de roostergas benadering. De stap van roostergas naar rooster Boltzmann en de afleiding van de rooster Boltzmann-methode uit de Boltzmann vergelijking beëindigen dit hoofdstuk. De intentie van dit tweede hoofstuk is om lezers van *buiten* de CFD - en rooster Boltzmann gemeenschap een begrip van de methode te geven.

In hoofdstuk drie worden de basisconcepten voor een efficiënte implementatie van de rooster Boltzmannmethode besproken. De noodzaak om ge-avanceerde algoritmen en datastructuren te gebruiken, teneinde de benodigde rekenprestaties te halen om realistische problemen aan te kunnen pakken, wordt geïllustreerd door een gedetailleerde prestatieanalyse, waarin een volle-matrix en een ijle-matrix rooster Boltzmann code worden vergeleken. Het punt van MPI parrallelisatie wordt kort besproken, en een introductie in de specifieke aanpak voor de visualisatie van HPC simulaties beëindigd dit hoofdstuk.

Hoofdstuk vier bevat gedetailleerde quantitatieve studies naar de vloeistofstroming in complexe geometrieën. Na een initieële validatie van de methode door een vergelijking met analytische oplossingen en andere numerieke resultaten, is het probleem van drukverlies in complexe geometrieën geanaliseerd. Gedetailleerde simulaties van stroming in poreuze media hebben aangetoond dat turtuositeit wordt overschat in de gebruikelijke capilaire theorieën, en de vervorming van vloeistofelementen is geïdentificeerd als een tweede bron van dissipatie, die in deze capillaire theorieën meestal wordt verwaarloosd.

Naast een bijdrage aan het onderzoek in de vloeistofmechanica was het de intentie van dit hoofdstuk om de rooster Boltzmannmethode te promoten als een efficiente en competatieve methode voor simulaties van complexe stromingen.

In hoofdstuk vijf worden uitbreidingen aan de stromingsimulator besproken, bedoeld voor het modeleren van multi-fysica fenomenen. Door relatief eenvoudige uitbreidingen van de lokale regels (gebruikmakend van de cellulaire automaat structuur van de rooster Boltzmannmethode) wordt aangetoond dat verrassend complexe structuren kunnen ontstaan op de macroschaal. Dit hoofdstuk illustreert de kracht van de rooster Boltzmannmethode voor simulaties van complexe stromingen en multi-fysica toepassingen, via qualitatieve en quantitatieve simulaties van verschillende onderzoeksvelden, zoals heterogene catalitische reacties, adsorptie en melk - en bloedstolling.

Het laatste hoofdstuk zes bevat een samenvatting van de rooster Boltzmann software ontwikkeling en geeft een aantal argumenten voor industrie en onderzoekers om realistische problemen met de rooster Boltzmannmethode aan te pakken. Een korte omschrijving van het huidige onderzoek van de auteur binnen een tweetal Europese projecten op het gebied van medische fysica besluit dit proefschrift.

Acknowledgments

My very special thanks goes to Peter Sloot, who kindly offered to be my promotor and gave me the opportunity to complete this work under his supervision. Very productive discussions sharpened the focus of this thesis and helped to structure the compilation of almost ten years of research in a (hopefully) reasonable way.

Alfons Hoekstra, my co-promotor (who did not give up motivating me to compile a thesis for submission at the University of Amsterdam) is an inspiring fellow researcher, who was incredibly helpful by giving important corrections and hints for improving this thesis.

Past Supervisors

Hans-Joachim Schellnuber (former: Interdisciplinary Centre for Marine and Environmental Sciences (ICBM), University of Oldenburg, now: Potsdam Institute for Climate Impact Research (PIK)) for introducing the Lattice-Gas and Lattice-Boltzmann methods to me (well against my original interests in nonlinear dynamics and self-organised criticality at that time) and supervising my first steps in scientific research.

Franz Durst (former: Institute of Fluid Mechanics (LSTM), University of Erlangen, now retired) for being bold enough to invest in Lattice-Boltzmann at a time when the method was almost unknown in the area of applied CFD, by offering me a research position at his institute and guiding my research for more than four years.

Michael Schäfer (former LSTM, now TU Darmstadt), Gunther Brenner (former LSTM, now TU Clausthal) and Michael Breuer (LSTM) for supervising my research at LSTM in the years 1994-1999.

Senior Colleagues

Markus Baum (former CCRLE, now DaimlerChrysler AG, Germany) who taught me the first steps in performance oriented implementation on super-computers, and who established the contact to NEC.

Bastien Chopard, whose very motivating COSMASE workshop and lectures in Geneva 1996 helped me significantly in understanding the concept behind Lattice-Gas and Lattice-Boltzmann at a time where textbooks were not yet available.

Sauro Succi, who convinced me to use the simpler lattice-BGK scheme instead of his own approach at a time when I was close to give up.

Manfred Krafczyk and Jonas Tölke, who are supervising probably one of the best and most productive teams of PhD students in the area of Lattice-Boltzmann, for a long-term co-operation and the possibility of many enlightening discussions with their group.

Li-Shi Luo, who never hesitated to boil down his concept of the theory to a level that even I could (at least partially) follow, and also for kind invitations to present our research at the international ICMMES conferences in Hong Kong (2005), Hampton (2006) and Munich (2007).

Guy Lonsdale for providing and protecting a niche for Lattice-Boltzmann CFD research at the NEC Laboratories Europe, St.Augustin.

Hanamura-san (former NEC Japan, now RIKEN), Takahara-san and Masuda-san from NEC Japan, who introduced me to the researchers of all major Japanese automotive companies, which gave a unique insight into the problems of industrial CFD application.

Rod Hose and Pat Lawford for introducing the field of Medical Physics to me and for establishing a fruitful research co-operation, full of critical discussions on the applicability and limitations of Lattice-Boltzmann.

Students

A major part of research work is done by the (PhD) students, who are not yet eaten up by a daily routine. I enjoyed being involved in supervising the work of:

Sigrid Schreiber, Stefan Braun, Francois D'Hospital, Murat Sabanca, Pascal Raud, Lilit Axner and Jan Linxweiler and particularly of the former students and now fellow researchers:

Thomas Zeiser, Peter Lammers and Sarah Harrison,¹⁴ with whom I have been enjoying years of productive co-operation, part of which contributed to this thesis.

And last but not least, I want to thank my colleagues at the NEC Laboratories Europe for a pleasant working environment and many interesting discussions.¹⁵

¹⁴My very special thanks to Sarah for a native-speaker's spell check of my thesis! All mistakes and typos still left are due to my ignorance considering her suggestions.

¹⁵It might well be noted that the coffee produced on second floor not always provided the minimum required concentration of caffeine - a complex-flow multi-physics problem which still has to be solved.

Publications

Submitted Papers

- J. Bernsdorf, Lattice-Boltzmann Simulation of Reactive Flow through a Porous Media Catalyst. In *Proceedings of the JSAE Symposium, Tokyo, Japan, December 14,* 2007, submitted 2007.
- D. J. W. Evans, P. V. Lawford, J. Gunn, D. Walker, D. R. Hose, R. H. Smallwood, B. Chopard, M. Krafczyk, J. Bernsdorf and A. G. Hoekstra. The application of multiscale modelling to the process of development and prevention of stenosis in a stented coronary artery. *Philosophical Transactions of the Royal Society A*, submitted 2007.
- J. Bernsdorf and D. Wang. Blood flow simulation in cerebral aneurysm: A lattice Boltzmann application in medical physics. In *Proceedings of the Parallel CFD 2007 Conference, Antalya, Turkey, May 16-20, 2007*, submitted 2007.
- L. Axner, J. Bernsdorf, T. Zeiser, P. Lammers, J. Linxweiler, and A. G. Hoekstra. Performance evaluation of a parallel sparse lattice Boltzmann solver. *J. Comput. Phys.*, submitted 2007.
- J. Bernsdorf and D. Wang. Non-Newtonian blood flow simulation in cerebral aneurysms. In *Proceedings of the ICMMES 2007 Conference, Munich, Germany, July 16-20, 2007*, submitted 2007.
- D. Wang and J. Bernsdorf. Lattice Boltzmann simulations of steady shear thinning blood flow with a Carreau-Yasuda model. *Proceedings of the Parallel CFD 2007 Conference, Antalya, Turkey, May 16-20, 2007*, submitted 2007.

Fully Reviewed Publications

- S. E. Harrison, J. Bernsdorf, D. R. Hose, and P. V. Lawford. A lattice Boltzmann framework for simulation of thrombogenesis. *Progress of CFD*, accepted for publication 2007.
- S. E. Harrison, S. M. Smith, J. Bernsdorf, D. R. Hose, and P. V. Lawford. Application and validation of the lattice Boltzmann method for modelling flow-related clotting. *Journal of Biomechanics*, in press 2007.
- S. E. Harrison, J. Bernsdorf, D. R. Hose, and P. V. Lawford. A lattice Boltzmann framework for simulation of thrombogenesis. *Progress of CFD*, accepted for publication 2007.

- S. E. Harrison, J. Bernsdorf, D. R. Hose, and P. V. Lawford. Development of a lattice Boltzmann framework for numerical simulation of thrombosis. *Int. Journal of Modern Physics C*, 18(4):483–491, 2007.
- J. Bernsdorf, S. E. Harrison, D. R. Hose, P. V. Lawford, and S. M. Smith. Concurrent numerical simulation of flow and blood clotting using the lattice Boltzmann technique. *Int. J. Bioinformatics Research and Applications*, 2(4):371–380, 2006.
- J. Bernsdorf, S. E. Harrison, S. M. Smith, P. V. Lawford, and D. R. Hose. A lattice-Boltzmann HPC application in medical physics. In M. Resch, Th. Bönisch, S. Tiyyagura, T. Furui, Y. Seo, and W. Bez, editors, *High Performance Computing on Vector Systems 2006*, Proceedings of the High Performance Computing Center Stuttgart, pages 121–132. Springer, 2006.
- J. Bernsdorf, S. E. Harrison, D. R. Hose, P. V. Lawford, and S. M. Smith. Numerical simulation of clotting processes: A lattice Boltzmann application in medical physics. *Mathematics and Computers in Simulation*, 72(2-6):89–92, 2006.
- T. Takei, J. Bernsdorf, N. Masuda, and H. Takahara. Lattice Boltzmann simulation and its concurrent visualization on the SX-6 supercomputer. In *HPCASIA '04: Proceedings of the High Performance Computing and Grid in Asia Pacific Region, Seventh International Conference on (HPCAsia'04)*, pages 212–219, Washington, DC, USA, 2004. IEEE Computer Society.
- J. Bernsdorf, U. Jaekel, T. Zeiser, T. Takei, H. Matsumoto, and K. Nishizawa. Lattice Boltzmann simulation and visualisation of adsorption processes in complex geometries. In P. M. A. Sloot, D. Abramson, A. V. Bogdanov, J. Dongarra, A. Y. Zomaya, and Y. E. Gorbachev, editors, *Computational Science - ICCS 2003, International Conference, Melbourne, Australia and St. Petersburg, Russia, June 2-4, 2003. Proceedings, Part I*, volume 2657 of *Lecture Notes in Computer Science*, pages 1054– 1061. Springer, 2003.
- J. Bernsdorf, U. Jaekel, T. Zeiser, S. Doi, T. Takei, H. Matsumoto, and K. Nishizawa. Numerical simuration of reaction-diffusion and adsorption processes in porous media using lattice Boltzmann methods with concurrent visualisation. In K. Matsuno, A. Ecer, N. Satofuka, J .Periaux, and P. Fox, editors, *Parallel Computational Fluid Dynamics 2002, Proceedings of the Parallel CFD 2002 Conference, Kansai Science City, Japan, May 20-22, 2002*, pages 233–240. Elsevier, 2003.
- G. Brenner, T. Zeiser, K. Beronov, P. Lammers and J. Bernsdorf: Lattice Boltzmann methods: High performance computing and engineering applications. In K. Matsuno, A. Ecer, N. Satofuka, J .Periaux, and P. Fox, editors, *Parallel Computational Fluid Dynamics 2002, Proceedings of the Parallel CFD 2002 Conference, Kansai Science City, Japan, May 20-22, 2002*, pages 3–12. Elsevier, 2003.
- T. Zeiser, M. Steven, H.-J. Freund, P. Lammers, G. Brenner, F. Durst, and J. Bernsdorf. Analysis of the flow field and pressure drop in fixed bed reactors with the help of lattice Boltzmann simulations. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 360(1792):507–520, 2002.

- T. Zeiser, H.-J. Freund, J. Bernsdorf, Y.-W. Li, P. Lammers, G. Brenner, and F. Durst. CFD calculations of flow, dispersion and chemical reactions in fixed-bed tubular reactors using the lattice Boltzmann method. In M. Breuer, F. Durst, and Ch. Zenger, editors, *High Performance Scientific and Engineering Computing: Proceedings of the 3rd International FORTWIHR Conference on HPSEC*, volume 21 of *Lecture Notes in Computational Science and Engineering*, pages 53–62. Springer, 2002.
- T. Zeiser, H.-J. Freund, J. Bernsdorf, P. Lammers, G. Brenner, and F. Durst. Detailed simulation of transport processes in reacting multi-species flows through complex geometries by means of the lattice Boltzmann method. In E. Krause and W. Jäger, editors, *High Performance Computing in Science and Engineering '01, Transactions of the High Performance Computing Center Stuttgart (HLRS) 2001*, pages 442–452. Springer, 2002.
- T. Zeiser, G. Brenner, J. Bernsdorf, P. Lammers, and F. Durst. Performance aspects of lattice Boltzmann methods for applications in chemical engineering. In C. B. Jenssen et al., editor, *Parallel Computational Fluid Dynamics 2000, Trends and Applications, Proceedings of the Parallel CFD 2000 Conference, May 22-25, Trondheim, Norway*, pages 407–414. Elsevier, 2001.
- J. Bernsdorf, T. Zeiser, P. Lammers, G. Brenner, and F. Durst. Perspectives of the lattice Boltzmann method for industrial applications. In C.B. Jenssen et al., editor, *Parallel Computational Fluid Dynamics 2000, Trends and Applications, Proceedings of the Parallel CFD 2000 Conference, May 22-25, Trondheim, Norway*, pages 367–373. Elsevier, 2001.
- J. Bernsdorf, G. Brenner, T. Zeiser, P. Lammers, and F. Durst. Numerical analysis of the pressure drop in porous media flow using the lattice Boltzmann computational technique. In N. Satofuka, editor, *Computational Fluid Dynamics 2000, Proceedings of the First International Conference on Computational Fluid Dynamics, Kyoto, Japan, 10-14 July 2000*, pages 493–498. Springer, 2001.
- T. Zeiser, P. Lammers, E. Klemm, Y. W. Li, J. Bernsdorf, and G. Brenner. CFDcalculation of flow, dispersion and reaction in a catalyst filled tube by the lattice Boltzmann method. *Chem. Engng. Sci*, 56(4):1697–1704, 2001.
- M. Breuer, J. Bernsdorf, T. Zeiser, and F. Durst. Accurate computations of the laminar flow past a square cylinder bases on two different methods: Lattice-Boltzmann and Finite-Volume. *Int. J. Heat and Fluid Flow*, 21(2):186–196, 2000.
- J. Bernsdorf, G. Brenner, and F. Durst. Numerical analysis of the pressure drop in porous media flow with lattice Boltzmann (BGK) automata. *Computer Physics Communications*, pages 247–255, 2000.
- J. Bernsdorf, F. Durst, and M. Schäfer. Comparison of cellular automata and finite volume techniques for simulation of incompressible flows in complex geometries. *Int. J. Numer. Met. Fluids*, 29:251–264, 1999.
- J. Bernsdorf, T. Zeiser, G. Brenner, and F. Durst. Simulation of a 2D channel flow around a square obstacle with lattice-Boltzmann (BGK)-automata. *Int. Journal of Modern Physics C*, 9(8):1129–1141, December 1998.

• J. Bernsdorf and M. Schäfer. Practical aspects of the simulation of viscous flow using lattice Boltzmann automata. In R. Helmig, W. Jäger, W. Kinzelback, P. Knabner, and G. Wittum, editors, *Notes on numerical fluid mechanics: Modeling and comutation in environmental sciences*, volume 59, pages 61–71. Vieweg, 1997.

Seminar papers / other publications (not fully reviewed)

- J. Bernsdorf, S. E. Harrison, D. R. Hose, P. V. Lawford, and S. M. Smith. Concurrent numerical simulation of flow and blood clotting using the lattice Boltzmann technique. In J. Ma and L. T. Yang, editors, *Proceedings of the 11th International Conference on Parallel and Distributed Systems: ICPADS 2005; Fukuoka, Japan, July 20-22, 2005*, volume II, pages 336–340, 2005.
- A. G. Hoekstra, B. Chopard, P. Lawford, D. R. Hose, M. Krafczyk, and J. Bernsdorf. Introducing complex automata for modelling multi-scale complex systems. In *Proceedings of the European Conference on Complex Systems ECCS 2006, European Complex Systems Society, Oxford, UK, September 2006*, 2006.
- K. Beronov, T. Zeiser, H. Freund, J. Bernsdorf, G. Brenner and F. Durst. Packed bed reactor flow statistics for a low tube to particle diameter ratio obtained from 3D Lattice Boltzmann simulations. In: R. Bennacer and A. A. Mohamed, editors, *Proceedings of the 1st International Conference on Applications of Porous Media, June 2-8, 2002, Jerba, Tunesia*, pages 1–13, 2002.
- J. Bernsdorf, G. Brenner, S. Doi, H. Matsumoto, K. Nishizawa, T. Takei, and T. Zeiser. Visualisation of reactive flows in environmental sciences and chemical engineering using rvslib and lattice Boltzmann flow solver. In *Proceedings of the First International Symposium on Advanced Fluid Information, Sendai, Japan, October 4-5, 2001*, pages 292–395, 2001.
- J. Bernsdorf. Numerische Simulation reaktiver Strömungen in komplexen Geometrien. *Jahresbericht AG STAB*, pages 208–209, 1997.
- J. Bernsdorf. Flow simulations using cellular automata lattice Boltzmann methods. Ch. Schaller and M. Brehm, editors, *Overview of Research projects on the Cray YM-P at the Leibniz-Rechenzentrum Mnchen*, pages 41–44, 1996.
- J. Bernsdorf. Numerische Simulation reaktiver Strömungen in komplexen Geometrien. *Jahresbericht AG STAB*, pages 195–196, 1996.
- J. Bernsdorf and M. Schäfer. Comparison of cellular automata and finite volume techniques for simulation of incompressible flow in porous media. *ERCOFTAC Bulletin* 28:24–27, 1996.