Fluid Dynamics and Ship Theory

TUHH

Masterarbeit

Development of Overset Strategies for LBM-Based Flow Solvers

Henrik Asmuth, B.Sc.

Matrikelnr.: 21045969

14. 10. 2016

Erstprüfer:
Prof. Dr.-Ing. Thomas Rung
Institut für Fluiddynamik und Schiffstheorie (M-8)

Zweitprüfer:
Prof. Dr.-Ing. Gerhard Schmitz
Institut für Technische Thermodynamik (M-11)

Betreuer:
Dr.-Ing Christian F. Janßen
Institut für Fluiddynamik und Schiffstheorie (M-8)

Ich, Henrik Asmuth, versichere an Eides statt, dass ich die vorliegende Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Die Arbeit wurde in dieser oder ähnlicher Form noch keiner Prüfungskommission vorgelegt.

Hamburg, den 14. 10. 2016

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Within the last two decades the Lattice Boltzmann Method (LBM) underwent tremendous developments. Emerging from the lattice gas cellular automata in the late 1980's (McNamara and Zanetti, 1988), it nowadays states a powerful alternative to classical computational fluid dynamic approaches directly solving the Navier-Stokes (NS) equations. Important steps towards its applicability in a wide range of technical and scientific applications were the introduction of multi-relaxation-time collision operators (Lallemand and Luo, 2000; d'Humières et al., 2002), pressure and velocity boundary conditions (Junk and Yang, 2009b) and higher-order no-slip boundary conditions (e.g. Lallemand and Luo, 2003), to name a few. Driving factors motivating this development are the high computational efficiency, the strict locality of the numerical scheme allowing for efficient parallelisation and the consideration of physical effects beyond the Navier-Stokes hydrodynamics. Both, in academia and industry, the LBM is now applied to solve single-phase, multi-phase and free-surface flows.

Many fluid dynamic problems, i.e. floating bodies, rotating machine parts or rising bubbles need to consider bodies that move relative to the fluid. Only in special cases the problem can be rearranged such that the body's motion is considered in the flow conditions while the object is not moving in the computational domain. The latter refers to classical wind tunnel experiments (and its numerical equivalents), e.g. a horizontally moving plane or a driving car. For more complex problems the motion of the object has to be incorporated directly in the simulation. Hence, the grid has to be updated according to the orientation of the moving body. In most Navier-Stokes finite volume codes this is done using mesh distortion techniques or moving overset grids, often referred to as Chimera grids (Thompson et al., 1999, chap. 37).

Due to the symmetry of the discrete velocity set of the Lattice Boltzmann equation (LBE), most LB codes utilise regular Cartesian grids. The underlying methods are thus usually not designated for any alteration of the grid. In order to maintain the structure of the grid and account for moving geometries, one approach is the application of fill algorithm switching the state of lattice nodes from solid to fluid and vice versa while tracking the actual boundary of the body and applying an interpolated bounce-back scheme (Lallemand and Luo, 2003). However, this method has two major drawbacks. First, the reinitialisation of fluid nodes using interpolations from neighbouring nodes imposes an error that shows in spurious pressure oscillations. Secondly, grid refinements that are often applied around moving (or stationary) bodies are not inherently moving with the body. This again can be overcome using adaptive mesh refinements (AMR), see e.g. Wood and Deiterding (2015). Such techniques, though, are rather complex to implement and not

suitable for hardware architectures like GPUs as utilised in the LB code ELBE (*efficient lattice Boltzmann environment*) and thus this work.

One possibility to allow for moving bodies with body fixed grid refinements and without the necessity to reinitialise fluid nodes is to apply moving overset grids, which are well established in the field of classical NS CFD, to the LBM. One, usually refined, grid including the geometry of the body is bound to the body following its position throughout a simulation. Within the moving grid the body is consequently stationary avoiding the above-mentioned problems of moving geometries in the LBM. Still, the lattice Boltzmann equation can be solved on regular Cartesian grids.

This work presents the development, implementation and testing of a moving overset grid approach for the LBM. The developed approach is implemented in the GPU-accelerated LBM environment ELBE developed at the Institute for Fluid Dynamics and Ship Theory (FDS) at Hamburg University of Technology.

The outline of the presented thesis reads as follows: Chapter 2 gives a brief introduction of the lattice Boltzmann method. Besides the fundamental theory the focus is laid on specific methods utilised in this work like the multiple-relaxation-time collision operator or body force formulations. In chapter 3 general approaches for moving grids in computational fluid dynamics are discussed and, subsequently, an outline of the most common techniques applied for the simulation of moving bodies in LBM frameworks is given. Chapter 4 presents the theoretical description of the moving overset approach developed in this work. Starting from stationary overset grids all necessary transformation and scaling procedures are outlined. The implementation of the approach is described in the subsequent chapter. Here, a description of the developed node search algorithms and their parallelisation within CUDA is given. Chapters 5.4 and 6 show the performed verification and validation studies on two and three dimensional grids. Finally, chapter 7 presents the conclusion and summarised the most important aspects to be improved as part of future implementations.

Chapter 2

The Lattice Boltzmann Method

Tackling the drawbacks of the lattice gas cellular automata (LGCA), as for instance the famous FHP model (Frisch et al., 1986), the initial approach of the Lattice Boltzmann Method was introduced by McNamara and Zanetti (1988). The method soon became its own discipline within the vast range of numerical models for the simulation of fluid dynamic problems. In this chapter the basics of the LBM are discussed, starting from the Boltzmann equation and discussing its discretisation in order to solve it numerically on a two- or three-dimensional lattice (sections 2.1 and 2.2). Furthermore, topics of particular relevance for this work are discussed in greater detail. Such are especially the moment space, moving boundary conditions and moving bodies and the application of body forces. For more information on LGCAs and the gas kinetic fundamentals of the Boltzmann equation see, for instance, Wolfram (1986), Succi (2001) or Mohammad (2011).

2.1 The Boltzmann Equation

The fundamental equation of the LBM is the Boltzmann equation, introduced by Ludwig Eduard Boltzmann in 1872. Boltzmann, the father of statistical mechanics, was the first to state a relation between the microscopic particle dynamics and macroscopic fluid properties such as density, viscosity or temperature.

As opposed to molecular dynamics considering the kinetics of every individual particle (a molecule or atom), the Boltzmann equation describes the evolution of particle distribution functions (PDFs) f. Here $f(t, \boldsymbol{x}, \boldsymbol{\xi})$ states the probability of finding a particle at time t at position $\boldsymbol{x}$ with velocity $\boldsymbol{\xi}$ (Sukop and Thorne, 2007). Boltzmann's approach is therefore classified as being mesoscopic, in contrast to the microscopic molecular dynamics or the macroscopic fluid description using the NS equations. $f(t, \boldsymbol{x}, \boldsymbol{\xi}) d\boldsymbol{x} d\boldsymbol{\xi}$ consequently yields the statistical number of particles at time t within an infinitesimal control volume $\boldsymbol{x} + d\boldsymbol{x}$ in the velocity range $\boldsymbol{\xi} + d\boldsymbol{\xi}$. The evolution of f can thereupon be deduced as

$$\frac{\partial f(t, \boldsymbol{x}, \boldsymbol{\xi})}{\partial t} + \boldsymbol{\xi} \frac{\partial f(t, \boldsymbol{x}, \boldsymbol{\xi})}{\partial \boldsymbol{x}} + \boldsymbol{F} \frac{\partial f(t, \boldsymbol{x}, \boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \Omega \quad , \tag{2.1}$$

the Boltzmann equation, where $\boldsymbol{F}$ is an external force and Ω the Collision operator. The three terms on the left-hand side, from left to right, describe the transient rate of change of f, the change due to convection and the change due to an external force. However, the latter will subsequently be neglected. Within the LBM it will be added as a final step after the collision, see section 2.5. The collision operator on the right-hand side expresses

the interaction of particles and is again dependent on f. Apart from the left-hand side containing partial derivatives of t, $\boldsymbol{x}$ and $\boldsymbol{\xi}$ the main problem in solving the LBE is the non-linear collision operator. It is therefore usually linearised. The most widely applied and probably most simple formulation is the one by Bhatnagar et al. (1954), further outlined in section 2.3.1.

2.2 The Discrete Boltzmann Equation

In order to solve the Boltzmann equation numerically, it is discretised by means of a two- or three-dimensional lattice of width Δx and a corresponding discrete velocity space. Possible locations of particles are thus restricted to the node positions of the lattice. Likewise, the possible velocities within a unit lattice cell are confined to the discrete directions $\boldsymbol{h}_i$ from one lattice node to its neighbours. Introducing the unit lattice velocity c, with

$$c = \frac{\Delta x}{\Delta t} \quad , \tag{2.2}$$

where Δt is the discrete time step, the velocity vectors within the lattice cell are given by $\boldsymbol{e}_i = c \cdot \boldsymbol{h}_i$. Their magnitude is thus also implicitly constrained by the lattice. Note that the entire discretisation underlies the assumption of a continuous fluid with Knudsen number $\mathrm{Kn} \ll 1$.

Given the aforementioned restrictions of the lattice, the continuous Boltzmann equation reduces to the discrete Boltzmann equation, expressing the evolution of f in n discrete lattice directions, with

$$\frac{\partial f_i(t, \boldsymbol{x})}{\partial t} + \boldsymbol{e}_i \frac{\partial f_i(t, \boldsymbol{x})}{\partial t} = \Omega_i \quad , i = 0, ..., n \quad . \tag{2.3}$$

The choice of the lattice both determines the range of physics to be incorporated and the numerical performance of the computation. The latter includes aspects such as accuracy, computational effort and stability. For the description of a lattice we shall follow the convention by Qian et al. (1992), namely DmQn, where m expresses the number of spacial dimensions and n the number of discrete lattice directions. Here, we shall only give a brief description of the lattices used within this work, the D2Q9 and the D3Q19. Note, however, that both more and less complex lattices are possible and also commonly used, see for instance Rubinstein and Luo (2008). More complex lattices generally allow for more physics to be incorporated . This again, is determined by the order of moment until which full isotropy is guaranteed.

The lattices used in this work are given in figure 2.1. The corresponding vector set for the D2Q9 velocity space of the lattice cell is given by

$$\begin{aligned} \boldsymbol{e}_{i;i=0,...,8} &= c \cdot \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & 1 & -1 \end{pmatrix} \\ &= \begin{pmatrix} f_R & f_E & f_N & f_W & f_S & f_{NE} & f_{NW} & f_{SE} & f_{SW} \end{pmatrix} \quad . \end{aligned} \tag{2.4}$$

For the three-dimensional D3Q19 lattice the velocity space reads

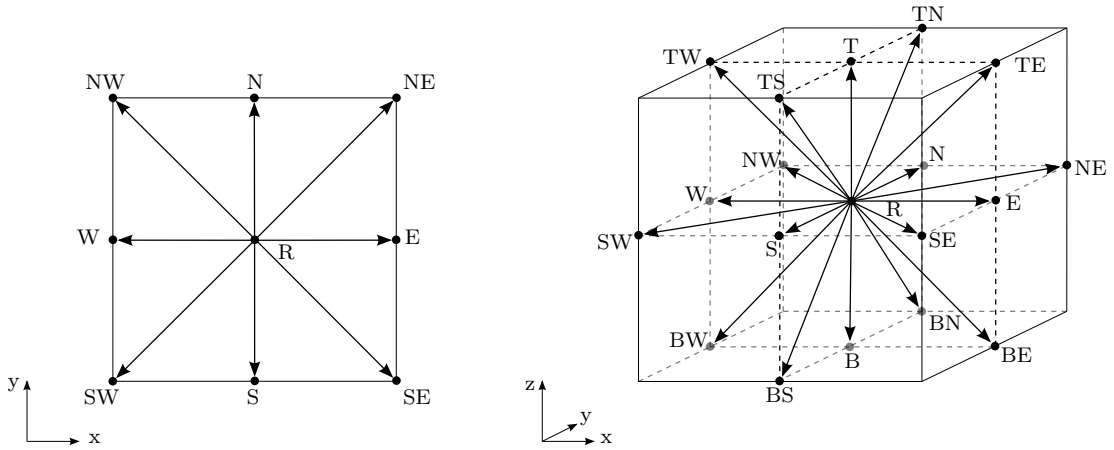


Figure 2.1: D2Q9 (left) and D3Q19 (right) lattice cells with all discrete velocity directions labelled in compass notation.

$$\boldsymbol{e}_{i;i=0,\ldots,18} = \tag{2.5}$$

$$c \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{pmatrix}$$

$$= (f_R\, f_E\, f_W\, f_S\, f_N\, f_T\, f_B\, f_{NE}\, f_{SW}\, f_{SE}\, f_{NW}\, f_{TE}\, f_{BW}\, f_{BE}\, f_{TW}\, f_{TN}\, f_{BS}\, f_{BN}\, f_{TS}) \quad .$$

Using an upwind finite differences (FD) scheme the discrete Boltzmann equation (2.3) can be discretised in space and time by

$$\frac{f_i(t+\Delta t, \boldsymbol{x}) - f_i(t, \boldsymbol{x})}{\Delta t} + c \cdot \frac{f_i(t+\Delta t, \boldsymbol{x} + \Delta t \cdot \boldsymbol{e}_i) - f_i(t+\Delta t, \boldsymbol{x})}{\Delta x} = \Omega_i \quad . \tag{2.6}$$

Since the lattice is usually equidistant and Cartesian, the choice of $c = 1$ simplifies equation 2.6 to

$$f_i(t+\Delta t, \boldsymbol{x} + \Delta t \cdot \boldsymbol{e}_i) - f_i(t, \boldsymbol{x}) = \Omega_i \quad , \tag{2.7}$$

the *Lattice*-Boltzmann equation. Often, the LBE is subsequently split into two separate equations describing a so called collision and propagation step, respectively. This not only facilitates the understanding of the actual physical meaning but also complies with the actual steps of the numerical procedure. First, the post-collision PDFs f_i' are computed based on the present distribution functions at one node and the chosen collision operator:

$$f_i'(t, \boldsymbol{x}) = f_i(t, \boldsymbol{x}) + \Omega_i \quad . \tag{2.8}$$

Subsequently they are propagated to their corresponding neighbour nodes, with

$$f_i(t+\Delta t, \boldsymbol{x} + \Delta t \cdot \boldsymbol{e}_i) = f_i'(t, \boldsymbol{x}) \quad . \tag{2.9}$$

With the velocity spaces used in this work the LBM can represent weakly compressible hydrodynamics. The macroscopic small density fluctuation ρ is accordingly given by the

zeroth moment of the PDFs, namely

$$\rho = \sum_{i=0}^{n} f_i \quad . \tag{2.10}$$

The first order moment yields the momentum

$$\rho_0 \boldsymbol{u}_i = \sum_{i=0}^{n} \boldsymbol{e}_i f_i \quad . \tag{2.11}$$

Using the lattice speed of sound, with $c_s = \frac{c}{\sqrt{3}}$ the pressure p is given by

$$p = \rho \cdot c_s^2 = \frac{1}{3}\rho \quad . \tag{2.12}$$

From a numerical point of view the described aspects of the discrete Boltzmann equation and its use within the LBM reveal its main benefits compared to solving the macroscopic NS directly. First of all, the collision step is strictly local and is thus predestinated for massive parallelisation. It is only dependent on the PDFs at each node. Every interaction with the neighbour nodes is restricted to the propagation step. Moreover, the propagation is a linear process. Hence, computationally, it boils down to copying PDFs from one node to another. And, additionally, the stress tensor including the pressure is explicitly given at each node following equation 2.12. Its determination can therefore be seen as a simple post-processing calculation. The implicit procedure of solving the Poisson-equation as in a NS framework thus also drops out.

2.3 Discrete Collision Operators

The complex nature of the collision operator necessitates its simplification in order to solve it numerically with reasonable effort. Within the last twenty years numerous collision operators were presented differing in complexity, numerical stability and computational cost. Herein an overview is given of the most simple one, the single-relaxation-time (SRT) collision operator by Bhatnagar, Gross and Krook (BGK, Bhatnagar et al., 1954) and the multiple-relaxation-time (MRT) collision operator. The latter will be used throughout all simulations of this work.

More complex collision operators like the cascading and the cumulant collison operator (Geier et al., 2006, 2009b) shall not be elaborated on. Details on their implementation and validation in ELBE are to be found in Gehrke (2015).

2.3.1 The Single-Relaxation-Time Model

In general the collision step describes a relaxation of the PDFs towards an equilibrium state, given by the equilibrium distribution functions f_i^{eq}. In the state of equilibrium the fluid is stress free and the Boltzmann equation reduces to $\Omega = 0$ which complies with a Maxwellian equilibrium given by

$$\boldsymbol{f}^{eq} = \frac{\rho}{(2\pi c_s^2)^{D/2}} \exp -\frac{(\boldsymbol{\xi} - \boldsymbol{u})^2}{2c_s^2} \quad , \tag{2.13}$$

with D denoting the spatial dimension of the underlying problem. For the BGK collision operator the Maxwellian equilibrium is approximated by a Taylor expansion of second order, with

$$f_i^{eq} = w_i \left(\rho + \rho_0 \left(3\frac{\boldsymbol{e}_i \cdot \boldsymbol{u}}{c^2} + \frac{9}{2}\frac{(\boldsymbol{e}_i \cdot \boldsymbol{u})^2}{c^4} - \frac{3}{2}\frac{\boldsymbol{u} \cdot \boldsymbol{u}}{c^2} \right) \right) \quad , \tag{2.14}$$

where ρ_0 is the reference density and w_i are the weights for the corresponding lattice nodes of the cell. For the D2Q9 these are given by

$$w_i = \begin{cases} \frac{4}{9} \ , & i = 0 \\ \frac{1}{9} \ , & i = 1, ..., 4 \\ \frac{1}{36}, & i = 5, ..., 8 \end{cases} \quad . \tag{2.15}$$

For the D3Q19 they read

$$w_i = \begin{cases} \frac{1}{3} \ , & i = 0 \\ \frac{1}{18}, & i = 1, ..., 6 \\ \frac{1}{36}, & i = 7, ..., 18 \end{cases} \tag{2.16}$$

The values of the weights result from a Chapman-Enskog-expansion of the LBE. They are chosen in order to preserve the consistency of the zeroth to second-order moments of the PDFs, namely the macroscopic density (equations 2.10), the momentum (2.11) and the stress tensor (introduced in equations 2.22 to 2.24 and, alternatively 4.5) with the incompressible NS equations. The latter consequently only applies under the condition of a low Knudsen number referring to the continuum assumption for a fluid. For further details and a derivation of the approximated equilibrium function the interested reader is referred to Succi (2001, chapter 5) and Krafczyk (2001).

The relaxation towards the above-mentioned equilibrium is expressed as

$$\Omega_i = -\frac{1}{\tau} \left(f_i - f_i^{eq} \right) \quad . \tag{2.17}$$

Note that all PDFs are relaxed with one single relaxation time τ reasoning the name of the collision operator. The difference $f_i - f_i^{eq}$ corresponds to the non-equilibrium part of the distribution function f_i^{neq} which again correlates with the viscous stress tensor, see section 2.3.2. The pre-collision non-equilibrium consequently states the result of external forces acting on the fluid. The relaxation time of the collision, τ, is given by

$$\tau = 3\frac{\nu}{c^2} + \frac{\Delta t}{2} \tag{2.18}$$

with ν being the lattice viscosity. Commonly the relaxation frequency ω, with

$$\omega = \frac{\Delta t}{\tau} = \frac{1}{\tau} \tag{2.19}$$

is used instead of τ. According to Qian et al. (1992) it should be chosen within the range of $\omega = [1, 2]$ in order to guarantee the numerical stability of the explicit procedure.

2.3.2 The Multiple-Relaxation-Time Model

In contrast to the SRT collision operator, the multiple-relaxation-time collision operator utilises a set of relaxation times in order to relaxate the statistical (velocity) moments of the PDFs. Subsets of the moments are thus relaxed with an individual relaxation time, allowing for a more precise adjustment of the relaxation itself.

Note that the general concept of moments and its mathematical description shall be outlined in greater detail in section 4.4.1 as it will play a major role in the transformation of PDFs between different grids. Here it will be confine to the transformation and relaxation and their physical interpretation within the LBM.

The transformation of discrete velocities and their corresponding particle distribution functions into the space of velocity moments is generally given by a matrix vector multiplication of the form $\boldsymbol{m} = \boldsymbol{M}\,\boldsymbol{f}$. It is thus linear and invertible implying that $\boldsymbol{f} = \boldsymbol{M}^{-1}\,\boldsymbol{m}$. Based on the m independent velocity vectors of a lattice and their corresponding particle distributions one can consequently construct m independent velocity moments. As opposed to particle distribution functions moments or '*modes*' thereof (see Lallemand and Luo, 2000) can provide a direct physical interpretation such as density or momentum. For each moment the relaxation time can consequently be chosen according to its hydrodynamic interpretation. This not only leads to an enhanced numerical stability but also overcomes several physical defects of the SRT such as a fixed Prandtl number and a fixed ratio of kinematic to bulk viscosity. Moreover, several models for multi-phase and free-surface flows are based on the moment formulation and would not be possible within the BGK framework. For further details on models based on the moment space and benefits with respect to stability see, e.g. d'Humières et al. (2002).

The moment space for the D2Q9 model used in this work and ELBE is given by the transformation as presented by Lallemand and Luo (2000), with

$$\boldsymbol{M} = \begin{pmatrix} c^0 \cdot (& 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ c^2 \cdot (& -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2) \\ c^4 \cdot (& 4 & 2 & 2 & 2 & 2 & -1 & -1 & -1 & -1) \\ c^1 \cdot (& 0 & 1 & 0 & -1 & 0 & 1 & -1 & 1 & -1) \\ c^3 \cdot (& 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1) \\ c^1 \cdot (& 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1) \\ c^3 \cdot (& 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1) \\ c^2 \cdot (& 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0) \\ c^2 \cdot (& 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1) \end{pmatrix} \quad . \tag{2.20}$$

The resulting moment vector reads

$$\boldsymbol{m} = \begin{pmatrix} \rho & e & \epsilon & j_x & q_x & j_y & q_y & p_{xx} & p_{xy} \end{pmatrix}^T \quad . \tag{2.21}$$

Here, ρ describes the density fluctuation as given in equation 2.10, j_x and j_y are the momentum components in x and y direction (corresponding to equation 2.11), respectively, and q_x and q_y are the x and y components of the energy flux. The fourth order moment ϵ corresponds to the energy dissipation. The second order moments e (density uncorrelated kinetic energy) and p_{xx} and p_{xy} correlate with the viscous symmetric stress tensor $\sigma_{\alpha\beta}$

(Geller, 2010). As a linear combination of moments of zeroth to third order it yields

$$\sigma_{11} = -\left(1 - \frac{\omega}{2}\right)\left(\frac{1}{3}\rho c^2 + \frac{1}{2}p_{xx} + \frac{1}{6}e - \rho_0 u_x^2\right) \tag{2.22}$$

$$\sigma_{22} = -\left(1 - \frac{\omega}{2}\right)\left(\frac{1}{3}\rho c^2 - \frac{1}{2}p_{xx} + \frac{1}{6}e - \rho_0 u_y^2\right) \tag{2.23}$$

$$\sigma_{12} = -\left(1 - \frac{\omega}{2}\right)\left(p_{xy} - \rho_0 u_x u_y\right) \tag{2.24}$$

In the moment space each moment is relaxed towards its equilibrium. The only exceptions state the density and the momentum being conserved quantities in the utilised athermal weakly compressible model. The collision of the non-conserved moments is given by

$$m_i = s_i \cdot (m_i - m_i^{eq}) \quad , i = 1, 2, 4, 6, 7, 8 \quad . \tag{2.25}$$

Grouping the values of s_i into subsets yields

$$s_{0,3,5} = 0 \quad , \quad s_1 = s_a \quad , \quad s_2 = s_b \quad , \quad s_{4,6} = s_c \quad , \quad s_{7,8} = s_d \quad . \tag{2.26}$$

According to Geller (2010) s_d is usually chosen as $s_d = \omega$, with ω being the relaxation frequency as used in the BGK model. The latter maintains consistency with respect to the kinematic viscosity related to s_d through the stress tensor (equation 2.22 to 2.24). The other non-zero relaxation rates can be chosen freely as $s_i = [\,0...2\,]$ in order to achieve optimal numerical stability. The precise values, however, are case-dependent and can not be determined a priori. In ELBE they are set as $s_a = s_b = s_c = 1$. Note that setting all $s_i = \omega$ we obtain the equivalent collision procedure as in the SRT model described in section 2.3.2. Another possibility states the so called two-relaxation-time model (TRT) by Ginzburg et al. (2008) where only two different relaxation times are applied to the odd and even moments, respectively.

Using the diagonal matrix $\boldsymbol{S} = diag(s_i)$ the entire collision step can be summarised as follows:

$$\Omega = \boldsymbol{M}^{-1}\,\boldsymbol{S}(\boldsymbol{M}\boldsymbol{f} - \boldsymbol{m}^{eq}) \quad , \tag{2.27}$$

where $\boldsymbol{m}^{eq}$ is the vector of equilibrium moments. Alike the equilibrium distribution functions, the equilibrium moments are functions of the reference density and the macroscopic fluid velocity. They can be obtained from

$$\boldsymbol{m}^{eq} = \begin{pmatrix} \rho \\ -2c^2\rho + 3\rho_0 \boldsymbol{u}\cdot\boldsymbol{u} \\ c^4\rho - 3c^2\rho_0 \boldsymbol{u}\cdot\boldsymbol{u} \\ \rho_0 u_x \\ -c^2\rho_0 u_y \\ \rho_0 u_x \\ -c^2\rho_0 u_y \\ \rho_0(u_x^2 + u_y^2) \\ \rho_0 u_x u_y \end{pmatrix} \quad . \tag{2.28}$$

For the D3Q19 the moment space is chosen according to Tölke et al. (2006). For the sake of brevity we shall not elaborate on the transformation matrix in more detail and only present the resulting moment vector:

$$\boldsymbol{m} = \tag{2.29}$$
$$\begin{pmatrix} \rho & e & \epsilon & j_x & q_x & j_y & q_y & j_z & q_z & 3p_{xx} & 3\pi_{xx} & p_{ww} & \pi_{ww} & p_{xy} & p_{yz} & p_{xz} & m_x & m_y & m_z \end{pmatrix}^T .$$

Compared to the D2Q9 we obtain the additional momentum and energy flux components j_z and q_z as well as additional components of the stress tensor p_{xy}, p_{yz} and p_{xz}. This leaves us with five further moments, namely $3\pi_{xx}$, π_{ww} and $m_{x,y,z}$ that do not refer to a physical interpretation in the context of the incompressible Navier-Stokes equations.

2.4 Boundary Conditions

When solving the LBE, the nodes at the boundary of a domain obviously lack neighbour nodes to receive PDFs from the respective lattice direction. The number of missing nodes certainly depends on the dimensionality of the grid and the position of the node at the boundary (edge, face or vertex). For the reconstruction of the incoming distribution functions and in order to apply certain flow conditions, boundary conditions are inevitable. Macroscopic quantities, as for instance a zero normal velocity along a wall or a specific inflow velocity can not be applied directly but require a conversion into PDFs. In what follows a description is given of the most important boundary conditions of the LBM used in this work. Please note that for each of the given boundary types numerous other approaches are to be found in the literature.

2.4.1 Inlet and Velocity Boundary Conditions

The application of a distinct inflow or far field velocity condition can be performed by directly prescribing the equilibrium of velocity $\boldsymbol{u}_b$ and density ρ_b at the boundary nodes following equation 2.14. The prescribed inflow is consequently stress free and thus only physically correct if the gradient of the velocity tangential to the boundary is set as zero. Otherwise a physical flow field will only be achieved a certain distance downstream of the boundary.

The problem can be overcome by applying more complex boundary conditions including the prescription of non-equilibria , see e.g. Zou and He (1997) or Guo et al. (2002b). Alternatively the boundary can be coupled to a co-simulation with periodic boundary conditions, so to speak, preparing a physical velocity field. Gehrke (2015) shows that this method can produce excellent results especially for highly turbulent inflow conditions that are often desired for LES simulations.

2.4.2 Outled and Pressure Boundary Conditions

Equivalent to the method described in section 2.4.1 equilibrium distribution functions can be used to define a certain outlet boundary condition. The drawbacks thereof are

of course similar to those given for the inlet. Janßen (2010) therefore suggests to use an extrapolation scheme in order to obtain the PDFs at the boundary.

Alternatively, we can specify a certain pressure at the outlet using the relation given in equation 2.10. Körner et al. (2005) present a corresponding method which, for the sake of brevity, shall not be elaborated on in this section.

2.4.3 No-slip Boundary Conditions

Along solid boundaries usually a Dirichlet condition is applied demanding the velocity tangential to the wall being equal to the velocity of the wall and that the velocity normal to the wall is set as zero. Again, the implementation thereof needs to be performed at the level of the particle distribution functions, namely using a so called bounce-back scheme. The concept provides that components of PDFs normal to the wall are reflected at the wall while tangential components are inverted. The most elementary bounce-back method is the Simple-Bounce-Back scheme (SBB, Lallemand and Luo, 2003). For distributions propagating towards the wall it yields

$$f_I(t, A) = f_i(t, S) - 6\rho_0 w_i \frac{\boldsymbol{e}_i \cdot \boldsymbol{u}_B}{c^2} \quad . \tag{2.30}$$

Here $f_I(t, A)$ denotes the PDF at the point A (see figure 2.2) in the inverse direction of the PDF at point S with $f_i(t, S)$ and $\boldsymbol{e}_I = -\boldsymbol{e}_i$. The second term on the right hand side considers the additional momentum acting on the particles in the case of the wall at position B moving with $\boldsymbol{u}_B$.

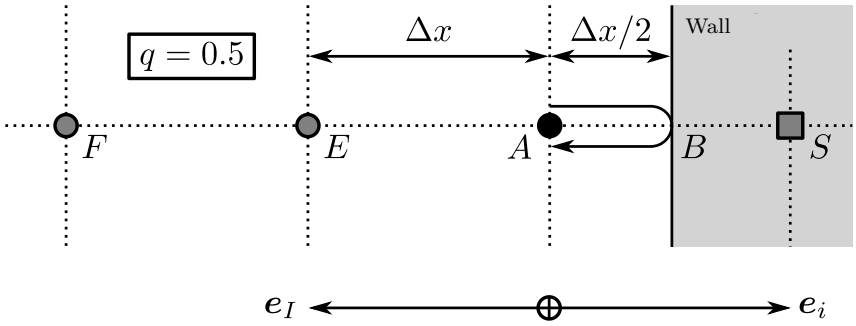


Figure 2.2: Simple-Bounce-Back scheme reflecting PDFs at boundary B. Source: Mierke (2015).

When applying the SBB, PDFs are usually saved in an additional layer of nodes around the actual fluid domain (here S). These *solid* or *obstacle* nodes only interact with the *fluid* nodes via the boundary condition.

With the normalised location of the wall

$$q = \frac{\overline{AB}}{\overline{AS}} \tag{2.31}$$

the scheme is second order accurate in space only for $q = 0.5$. For other wall locations it falls back to first order accuracy only (Bouzidi et al., 2001). The procedure consequently yields major accuracy issues for arbitrarily placed straight walls and particularly for curved boundaries. Interpolating schemes, however, can overcome this issue by incorporating

PDFs from the second and third row of nodes (E and F). Linear Interpolated Bounce-Back (LIBB) schemes, for instance, achieve second order accuracy for all q (Mei et al., 1999; Bouzidi et al., 2001; Lallemand and Luo, 2003). Here, the reflected PDFs are given by

$$f_I(t,A) = \begin{cases} 2q f_i(t,S) + (1-2q) f_i(t,A) - 6\rho_0 w_i \dfrac{e_i \cdot \boldsymbol{u}_B}{c^2} & , q \leq 0.5 \\ \dfrac{1}{2q} f_i(t,S) + \dfrac{2q-1}{2q} f_i(t,E) - \dfrac{3\rho_0 w_i}{q} \dfrac{\boldsymbol{e}_i \cdot \boldsymbol{u}_B}{c^2} & , q > 0.5 \end{cases} \quad . \tag{2.32}$$

The method obviously utilises the normalised wall location q. Hence it has to be computed additionally in every time step (in case the boundary is moving) for all fluid nodes neighbouring the boundary. For a detailed analysis of effective algorithms to compute q it is referred to Mierke (2015).

A further increase in accuracy is provided by the Quadratic Interpolated Bounce-Back (QIBB) also presented in Lallemand and Luo (2003). Yet investigations by Mierke (2015) show that results obtained from QIBB and LIBB only differ marginally. Simulations run as part of this work therefore apply the LIBB scheme for moving and non-moving boundaries.

2.5 Body Forces

In section 2.2 we neglected the forcing term given in the original Boltzmann equation. This complies with most numerical approaches which add an external body force as an additional term after the collision process. The method currently implemented in ELBE refers to the one presented in Luo (1993). Here the additional forcing term is given by

$$F_i = 3 w_i \rho \boldsymbol{e}_i \boldsymbol{F} \tag{2.33}$$

and added to the LBM as follows:

$$f_i(t + \Delta t, \boldsymbol{x} + \Delta t \cdot \boldsymbol{e}_i) - f_i(t, \boldsymbol{x}) = \Omega_i + F_i \quad . \tag{2.34}$$

The weights w_i agree with those given for the equilibrium distribution function (equation 2.14). As in many similar approaches, the simple underlying assumption is that the change of momentum induced by an external body force results in a post-collision momentum $\rho \boldsymbol{u}^{post}$ that is the average of the pre-collision momentum $\rho \boldsymbol{u}^{pre}$ and the momentum including the body force, yielding

$$\rho \boldsymbol{u}^{post} = \rho \boldsymbol{u}^{pre} + \frac{1}{2} \boldsymbol{F} \quad . \tag{2.35}$$

Following a Chapman-Enskog expansion it is shown by Buick and Greated (2000) as well as Guo et al. (2002a) that an an additional error term $\boldsymbol{j}_{err}$ is applied to the momentum equation of the NS given by

$$\boldsymbol{j}_{err} = \left(\tau - \frac{1}{1} \right) \Delta t \, \nabla \cdot (\boldsymbol{F}\,\boldsymbol{u} + \boldsymbol{u}\,\boldsymbol{F}) \quad . \tag{2.36}$$

This only vanishes for the special case of $\boldsymbol{F}$ being parallel to $\boldsymbol{u}$ and the non-linear term of the NS being zero. For most cases, and especially those shown in this work, this eventually does not persist and should be considered in the application of this approach. More suitable forcing representations as outlined in the aforementioned references might thus be worth considering in future approaches.

2.6 Normalisation to Lattice Units

Typically the LBE is solved in a dimensionless form. All physical quantities given in SI units are therefore nondimensionalised with respect to the characteristic lattice spacing Δx in [m] and the lattice speed of sound c_s in $[\mathrm{m\,s^{-1}}]$. Also note that the dimensionless units are often referred to as lattice units (lu).

The lattice spacing can straightforwardly be obtained from the number of lattice nodes n in one direction, here x, with

$$\Delta x = \frac{L_x}{n_x - 1} \quad , \tag{2.37}$$

where L_x is the domain length in the x direction. The lattice time step is hence given by

$$\Delta t = \frac{\Delta x}{c} = \frac{\Delta x}{c_s\,\mathrm{Ma}} \quad . \tag{2.38}$$

Using Δx and Δt as well as ρ_0, the normalisation factors for all relevant properties can easily be deduced. A summary of the most common quantities is given in table 2.1. Following the classical dimensional analysis any other quantity can of course be nondimensionalised accordingly.

Table 2.1: Normalisation factors N_ϕ for common physical units ϕ with $\phi_{lu} = N_\phi\,\phi_{\mathrm{SI}}$

Quantity	SI unit	Normalisation factor	
Velocity	$\mathrm{m\,s^{-1}}$	$(c_s\,\mathrm{Ma})/u_{\mathrm{SI}}$	with the reference velocity u_{SI}
Length	m	$n_x - 1/L_x$	with the reference length L_x
Time	s	N_u/N_x	
Density	$\mathrm{kg\,m^{-3}}$	$1/\rho_0$	with the reference density ρ_0
Force	$\mathrm{kg\,m\,s^{-2}}$	$N_\rho\,N_x^{D-1}/N_t^2$	
Pressure	$\mathrm{kg\,m^{-2}\,s^{-2}}$	N_F/N_x^{D-1}	

Chapter 3

Simulation of Flows around Moving Bodies

The interaction of fluids with moving bodies can be found in everyday life and in many technical applications. It can be characterised in many ways showing the diversity of possible interaction types. For the simulation of fluids in conjunction with a moving body such a classification is crucial in order to chose an appropriate numerical method.

First it needs to be determine whether the interaction of fluid and body is bi- or monodirectional. If the influence of the body on the fluid can be neglected the flow can be simulated solely. The resulting forces on a body can then be computed and integrated in the post-processing in order to obtain, e.g. the trajectories of the body. This applies, for instance, to dilute particle-laden flows with particle volume fractions of less than 10^{-6} (Lun et al., 1984). If the body's presence noticeably influences the fluid, as in most technical applications, it has to be accounted for in the grid of the simulation, including appropriate boundary conditions. Still, the coupling can be monodirectional given that the motion of the body is fully prescribed. In the case of a bidirectional coupling, both the motion (and/or deformation) of the body caused by the fluid as well as the bodies' influence on the fluid need to be solved for. Depending on the numerical method the two solvers can be coupled explicitly or implicitly. Furthermore, only certain degrees of freedom of the body can be permitted, as with wind turbines only rotating around one axis but sill necessitating a bidirectional coupling for this rotation.

Except for the first, above-mentioned case the geometry of the body needs to be represented by the grid. Hereby, the challenge is to update the grid in order to account for the changing orientation of the body. Within the field of CFD several methods emerged in order to cope with the stated problem. In this chapter an overview is given of methods commonly applied in NS finite volume or difference methods as well as the most common methods used in LBM codes.

3.1 Grid Adaptation Techniques

With the term *grid adaptation*, herein, techniques are summarised that change the initial structure of the grid. In order to follow the temporal change of orientation or shape of a body the location of grid points, their connectivity or a combination of the two is altered. As will be discussed in section 3.2, this contrasts strongly to moving overset grids where the grid structure within the stationary and the moving grid, respectively, does not

change.

3.1.1 Arbitrary Langrangian-Eulerian Methods

In the context of fluid dynamics, arbitrary Langragian-Eulerian methods (ALE) are a widespread approach in both commercial and academic codes to adapt the fluid domain to geometrical changes (Hadžić, 2006), see figure 3.1. As the name suggests, the method combines aspects of the particle-fixed Langrangian view-point and the spatially fixed Eulerian view-point.

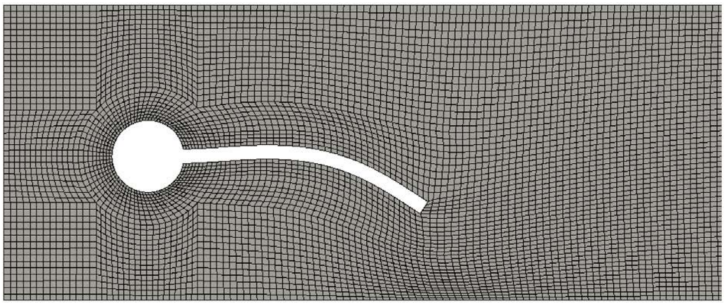


Figure 3.1: Exemplary distorted grid using the ALE method to follow the motion of a flexible trailing structure. Source: Miller et al. (2014).

The standard FD or FV approach for fluid dynamics is purely Eulerian. Momentum, mass and energy of a continuous fluid is examined by means of fixed control volumes represented by nodes or cells of a fixed grid. For instance, the momentum of an incompressible fluid within a control volume can then be given by the standard NS formulation

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nu \nabla^2 \boldsymbol{u} + S \quad , \tag{3.1}$$

where $\boldsymbol{u}$ denotes the absolute fluid velocity given in a fixed reference frame K and S summarises possible internal source or sink terms. In ALE formulations the nodes of a grid can move arbitrarily, meaning irrespective of the fluid particle motion. Still, the grid motion usually does not comply with the particle motion of the fluid. The mathematical description of the fluid thus remains Eulerian within the moving frame of reference of the grid resulting in a hybrid Langragian-Eulerian description. For the momentum equation 3.1 the fluid velocity $\boldsymbol{u}$ is therefore replaced by $\boldsymbol{u}'$ given in the moving frame of reference of the control volume K'. Furthermore, the convective velocity becomes $\hat{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}'$ resulting in

$$\frac{\partial \boldsymbol{u}'}{\partial t} + (\hat{\boldsymbol{u}} \cdot \nabla)\boldsymbol{u}' = \nu \nabla^2 \boldsymbol{u}' + S \quad . \tag{3.2}$$

According to Donea et al. (2004), most ALE methods can be seen as *steady* or *quasistatic*. Mesh accelerations and associated inertial forces acting on the fluid within the moving grid are therefore zero or small compared to other forces.

When following a moving body, technically, the only constraint on the grid distortion is given by the motion of the geometry of the body itself. Mesh quality, represented by the skewness and aspect ratio of a control volume, however is of major importance for

the accuracy of any numerical scheme. Grid regularisation and smoothing methods in order to control the so called *grid distortion* (Thompson et al., 1999) resulting from ALE methods can therefore be seen as a topic itself that shall not be further outlined in this work.

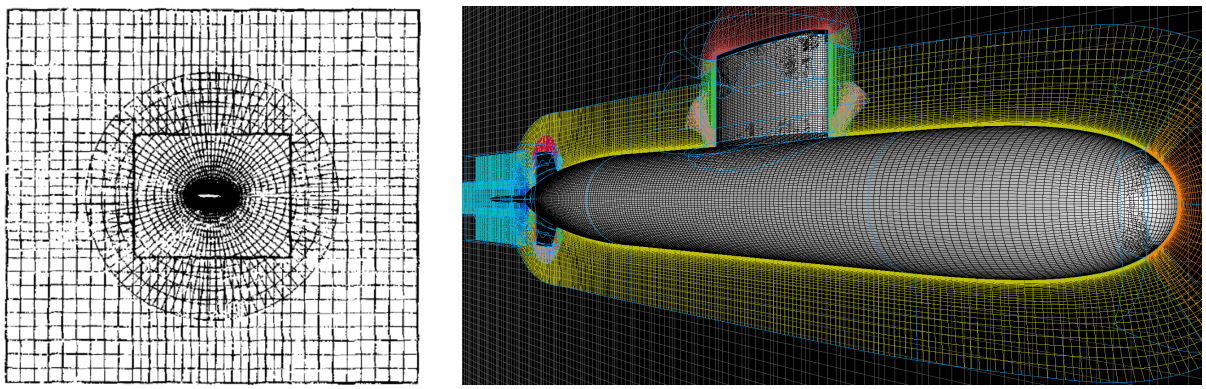
3.1.2 Static Regridding Methods

Regridding techniques create new grids during run-time in order to adapt to temporally changing requirements on the grid. Such are for instance changing refinements or moving geometries. The term static hereby denotes that the grid itself is not moving as opposed to ALE methods (Thompson et al., 1999). It becomes obvious that such methods can cope with more severe motions or deformations of bodies than methods that distort a given initial grid without changing the number and connectivity of nodes. Yet one major drawback is the interpolation of all fluid quantities from the old to the new grid and inherent inaccuracies thereof (Arney and Flaherty, 1990). Furthermore, following a moving geometry requires very small time steps compared to other methods.

For many applications a combination of the two methods presented here thus states an optimal set-up.

3.2 Overset Grids

Originally, the overset grid method was introduced to create composite grids. First presented by Atta (1981) the method enabled the connection of an O-type grid fitted to an airfoil and a Cartesian grid used for the outer subdomain, see figure 3.2a. The use of overset grids for moving subdomains states a more recent development.



(a) **(b)**

Figure 3.2: Examples of overset grids: (a) Overlapping of two-dimensional O-type and Cartesian grid. Source: Benek and Steger (1983). (b) Complex overset grid comprising multiple overlapping structured subgrids. Source: Sideroff (2015).

Generally speaking the overset method allows for the arbitrary combination of different structured grid types, each optimal for the requirements of a specific area of the computational domain. Unstructured grids or largely deformed structured grids can thus be

avoided when gridding complex geometries (Benek and Steger, 1983; Basso and Azevedo, 2004). Note, however, that more recently overset methods are also being applied to unstructured grids, see for instance Nakahashi et al. (2000) or Pandya et al. (2005).

3.2.1 Hole-cutting and Interpolation

Here a brief description is provided of the steps performed in the overset method for FD or FV schemes. For more details on the individual steps, see, e.g. Skillen (2011): The central aspect of overset grids is to pass fluid properties of two overlaying grids via interpolation from one grid to another in the fringe region. Donor and receptor nodes or cells therefore need to be identified on both grids alongside with deactivating nodes that are redundant for the computation due to overlaying other subgrids. The entire procedure is often referred to as *hole-cutting*. Note that for the sake of brevity it will only be referred to nodes in the following implying cells with respect to FV schemes.

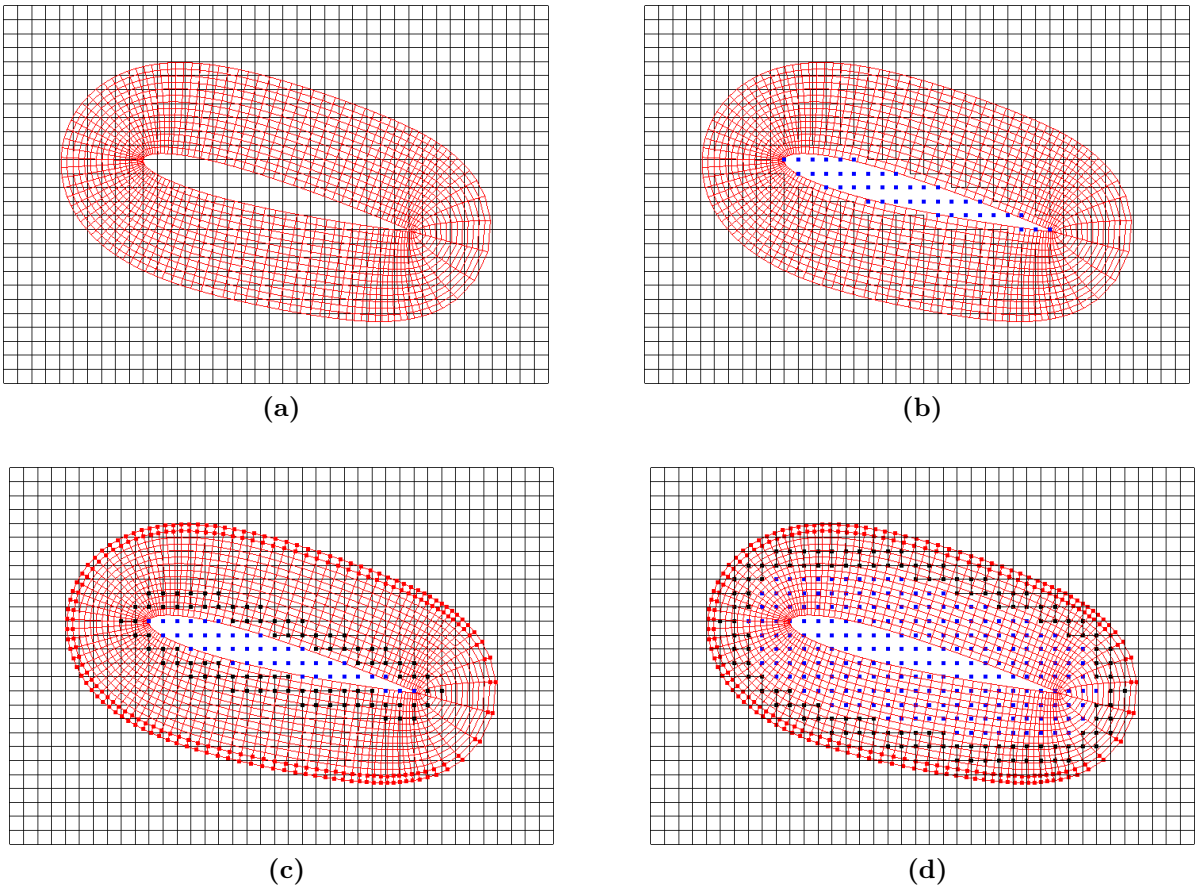


Figure 3.3: Schematic steps of the hole-cutting procedure with deactivated nodes marked in blue, receptor nodes of the background grid marked in black and receptor nodes of the refined body-fitted grid marked in red. Source: Celeritas Simulation Technology (2011)

Typically we hereby follow a hierarchical structure starting from the least prominent subgrid, usually a coarse background grid referring to the black Cartesian grid in figure 3.3. First of all, nodes intersecting with an overlying geometry are deactivated (see blue nodes,

figure 3.3b). The enclosing nodes on the subgrid are subsequently marked as preliminary fringe nodes (see black nodes, figure 3.3c). Also, the receptor nodes of the most prominent grid are marked (red nodes, figure 3.3c). These are typically the outermost nodes of the referring subdomain in order to fully utilise the refinement area. Subsequently, the inner area of deactivated nodes is expanded outwards whilst updating the enclosing nodes in order to maintain an intact fringe. The procedure is terminated with the fringe bordering the overlying receptor nodes (see figure 3.3d). During runtime the resulting black and red nodes, respectively, will receive fluid properties from the other subdomain resulting in a bidirectional coupling. For each receptor node the enclosing nodes in the other subdomain (donor nodes) thus have to be found in order to interpolate all required fluid properties to the other domain. Please note, the procedure also has to guarantee that each node may only serve as a receptor or donor. Combining the two may lead to numerical instabilities. Furthermore we should bare in mind that the entire hole-cutting procedure has to be repeated in each time step if the overlying grid is moving.

It becomes obvious that the receptor nodes on both grids also mark a boundary of the respective subdomain. Interpolating fluid properties from one grid to the other can thus also be seen as a dynamic boundary condition with the boundary values being provided by the other grid and not set a priori. Research on hole-cutting procedures mostly aims at optimisations with respect to computational efficiency (Pandya et al., 2005; Noack, 2005; Hadžić, 2006; Skillen, 2011). When it comes to interpolation schemes, numerical accuracy is the main concern. The most simple approach is a linear interpolation or to be precise a bi- or trilinear interpolation used for two or three dimensional grids, respectively. Linear interpolations, however, are only first order accurate in space and therefore inherently non-conservative. Various higher-order but still non-conservative schemes have been proposed to minimise interpolation errors. See for instance Henshaw (1994) proposing a fourth-order accurate scheme. Moreover, approaches for conservative or semi-conservative interface treatments (e.g. Tang et al., 2003) are to be found in the literature (for a detailed overview, see Skillen, 2011). To sum up, the choice of interpolation scheme remains a widely discussed topic led by the everlasting compromise of numerical accuracy and computational effort. After all, every scheme is conservative in the limit of a zero grid spacing, explaining the large discrepancies reported for the accuracy of the same interpolation scheme.

3.2.2 Moving Overset Grids

The overset method provides a straightforward way to use multiple interconnected grids in one computational set-up. Hence, arbitrary motions of a body represented in one subdomain can be performed without altering the structure of any grid. The body-bound grid simply moves with the body, then again resulting in a stationary body in the local reference frame of the referring subdomain. Even for large motions no regridding is required. The main additional computational effort therefore becomes the hole-cutting procedure that has to be repeated in every time-step.

One additional issue arising when moving an overset grid is the reinitialisation of inactive nodes (Hadžić, 2006). At reactivated nodes fluid properties of the last time step need to be known for the treatment of the unsteady terms of the Navier-Stokes equations.

These can be obtained using inter- or extrapolation from neighbouring active nodes, again implying the application of non-conservative schemes and inaccuracies thereof.

Computing within a moving reference frame of course implies further changes in the governing equations that were partly outlined in section 3.1.1. Note that a more detailed description will be given in chapter 4.

3.3 Moving Bodies in Lattice Boltzmann Methods

As outlined in chapter 2, the LBM is mostly applied on equidistant Cartesian grids. Except for local refinements which, again, are equidistant and Cartesian the utilised grids are not designated for any alteration of spacing or connectivity. To the author's knowledge grid adaptation techniques as outlined in section 3.1 are thus not commonly applied in LBM codes.

For the representation of moving bodies the most common methods applied in the context of the LBM are immersed boundary conditions and node updating in combination with bounce-back type boundary conditions. In the following the latter will be referred to as bounce-back methods only in compliance with most references (e.g., Lallemand and Luo, 2003; Peng and Luo, 2008; Rohde et al., 2008). Yet, it should be emphasised that the actual representation of the moving body is achieved by updating nodes from solid to fluid and vice versa in order to follow the geometry.

3.3.1 Immersed Boundary Conditions

In the immersed boundary condition the boundary of a moving body is discretised by a set of marker points. These can move arbitrarily within the fluid domain with fluid residing in and outside of the boundary. Referring to Nishida and Meichin (2012) the classical immersed boundary approach applies a simple body force $\boldsymbol{f}_{IB}$ to the lattice nodes near the virtual boundary represented by the markers. The momentum equation then yields

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nabla^2 \boldsymbol{u} - \nabla p + \boldsymbol{f}_{IB} \quad . \tag{3.3}$$

Using a forward Euler scheme $\boldsymbol{f}_{IB}$ can be determined as

$$\boldsymbol{f}_{IB} = \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \nabla^2 \boldsymbol{u} + \nabla p\right]^n + \frac{1}{\Delta t}\left(\boldsymbol{U}^{n+1} - \boldsymbol{u}^n\right) \quad , \tag{3.4}$$

where $\boldsymbol{U}$ denotes the velocity of the boundary marker. Due to spurious pressure oscillations and other accuracy issues numerous advancements of the immersed boundary conditions for the LBM are to be found in the literature and are being applied for complex moving geometries, see for instance Mittal and Iaccarino (2005). According to Peng and Luo (2008), however, (interpolated) bounce-back methods generally give more accurate results with the same computational effort. The simplicity and associated easy implementation of immersed boundary conditions can therefore be stated as their main advantage.

3.3.2 Bounce-back Boundary Conditions

In section 2.4.3 we discussed the basic concept of bounce-back schemes and the possibility to include a boundary velocity in the formulation (see equations 2.30 and 2.32). In ELBE and many other reported codes (e.g., Lallemand and Luo, 2003; Caiazzo, 2008; Geller, 2010) bounce-back methods are used to model the moving boundary while the geometry itself is represented by nodes flagged as *solid* referring to section 2.4.3. As mentioned above, the motion of the geometry necessitates that node states are switched during runtime.

The change from *fluid* to *solid* generally does not state a problem. More problematic is the opposite direction. It relates to the same issue when reactivating fluid nodes in the cut-out region of moving overset grids, outlined in section 3.2.2: The reinitialisation of fluid properties.

A conservative method for the reinitialisation is proposed by Mei et al. (2006). They suggest a local Poisson-like iteration solving the LBM at the new fluid nodes until convergence is achieved. Hereby the PDFs of the neighbour nodes remain constant and propagate to the new fluid node at every local collision step. As shown by Geller (2010) the method provides excellent results. Yet it is has major efficiency drawbacks due to its implicit nature and is thus not suitable for high-performance applications like ELBE.

Alternatively, the macroscopic properties $\boldsymbol{u}$ and ρ can be extrapolated from the neighbour nodes, see figure 3.4. The PDFs can subsequently be initialised using the correspond-

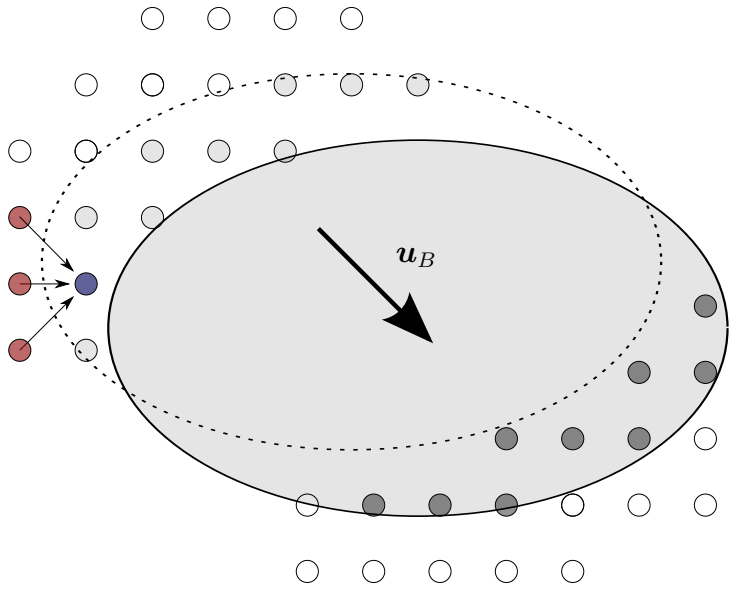


Figure 3.4: Schematic drawing of node updating for a moving body with velocity $\boldsymbol{u}_B$ in the LBM. Nodes switching from *solid* to *fluid* are marken in light grey, nodes switching from *fluid* to *solid* are marked in dark grey. Nodes used for the extrapolation of one exemplary node (blue) are marked in red.

ing equilibrium distribution $f^{eq}(\boldsymbol{u}, \rho)$. Obviously, this method is not mass conserving and disturbances of the flow field can be expected. Often these are reported as spurious pressure oscillations. As an example, simulation results from Lallemand and Luo (2003) are given in figure 3.5. The described phenomenon obviously states a major drawback of this method. Also, the severity of the erroneous fluctuations is dependent on the spacing of

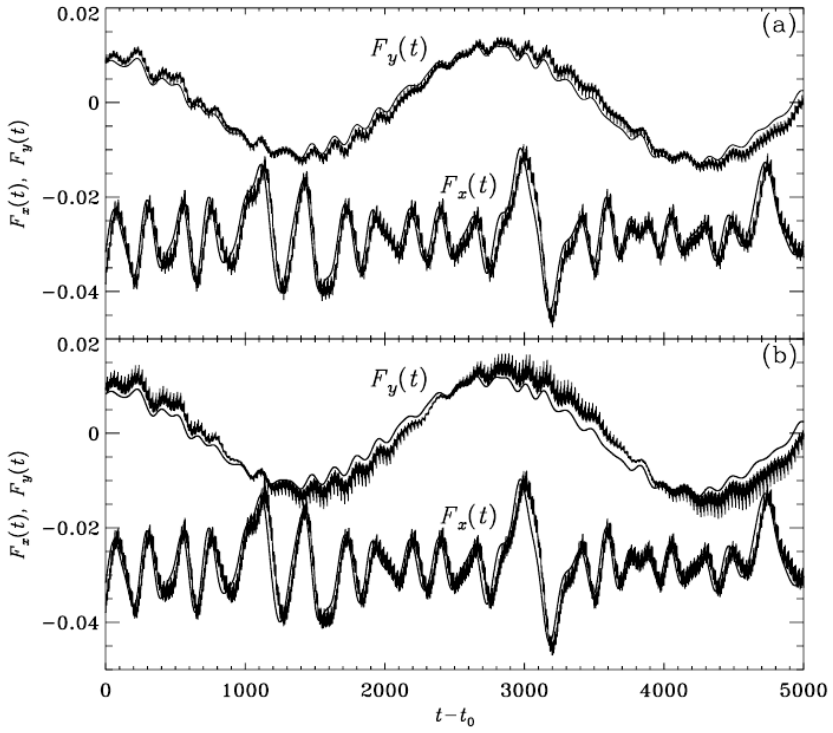


Figure 3.5: Example of spurious pressure oscillations due to node initialisation observable as oscillations of the overall lift F_y and drag force F_x acting on a cylinder. The smooth lines represent the forces of a stationary cylinder at Re = 200. The fluctuating lines refer to a horizontally moving cylinder with the same Reynolds number. (a) is obtained using the extrapolation scheme described here, (b) refers to a second order extrapolation of the non-equilibrium PDFs. Source: Lallemand and Luo (2003).

the grid and the order of the utilised extrapolation methods. Even though higher-order extrapolation schemes, as shown in figure 3.5, can reduce the fluctuations, the problem can not be fully overcome within the limits of acceptable computational effort. The describe problematic thus highlights the main motivation for the use of moving overset grids in the LBM.

Chapter 4

A Lattice Boltzmann Approach for Moving Overset Grids

For the representation of no-slip walls, stationary or moving, bounce-back schemes prove to be the method of choice for many applications. On the contrary, the overall suitability of node switching when moving a geometry may be doubted due to the accuracy issues outlined in section 3.3.2. Moreover, it is often desirable to refine the fluid domain around moving bodies. A common technique for this purpose are adaptive mesh refinements (AMR). They dynamically refine the fluid domain following certain criteria such as vorticity. Applications thereof range from microfluidic multiphase flows (Tölke et al., 2006; Yu and Fan, 2009) to large-scale technical applications like wind turbines (Wood and Deiterding, 2015). The dynamic data structure of AMR methods, though, makes them infeasible for GPU architectures (Valero-Lara and Jansson, 2015). And, after all, they can only reduce the interpolation errors of fluid node reeinitialisations as they locally decrease the node spacing.

Referring to section 3.2 it is revealed that moving overset grids combine several concepts that can overcome the above-mentioned issues. First, a body can be represented in a local reference frame. Within this reference frame (the moving grid) the body is stationary. Hence, nodes do not have to switch their states in order to follow its geometry. Second, a refinement around the body can be defined in the moving grid as well. It consequently moves automatically with the body. Moving overset grids, well established in conventional NS solvers, are thus promising also for LBM applications.

At first, this chapter shall give an overview of the state of the art of overset grids in the LBM. Subsequently the mathematical and numerical concepts of a moving overset grid strategy for the LBM are presented.

4.1 State of the Art

Technically the cell-vertex refinement method by Filippova and Hänel (1998), frequently applied for local refinements of fluid domains in the LBM, comprises all the characteristics of classical overset approaches. Interestingly it is usually not referred to as such. Common terms in the LBM community are *multi-domain* (Lagrava et al., 2012) or *embedded grid* (Chen et al., 2006) approach. Compared to classical overset approaches the only difference is the coupling of two identically structured grids, namely two Cartesian equidistant ones. Apart from that features like point-wise interpolation and hole-cutting procedures are

very much alike. In the following section it will be elaborated further on this method as it states a fundamental part of this work. In addition to that it is referred to Gehrke (2015) at this point. As opposed to most publications the author provides a transformation procedure to connect two equidistant Cartesian grids that are not aligned. Hereby the LBM is solved in each local frame of reference with one being rotated with respect to the other. Also, a qualitative analysis on a D2Q9 lattice is presented serving as a first prove of concept. The concept can therefore be seen as a LBM overset method for arbitrarily oriented, yet stationary grids.

To the author's knowledge moving overset grids for LBM have not been reported until now. For moving bodies it is usually referred to methods described in section 3.3. The only actual moving (rotating) grid approach is presented by Li (2011) and Zhang et al. (2015). The method described by the authors solves the LBM in a local, moving reference frame. The coupling of the rotating grid and the stationary grid is performed via interpolation of the macroscopic velocity $\boldsymbol{u}$ and density ρ to so called *surfels* with no necessary overlap. At the receiving *surfel* PDFs are reconstructed as $f^{eq}(\boldsymbol{u}, \rho)$. The continuity of the stress tensor along the interface is thus obviously violated. All in all the method is referred to as a *sliding mesh* and limited to one rotational degree of freedom of the moving mesh. Also note that, to the author's knowledge it is only being applied in the commercial LBM code PowerFLOW (Exa Corporation, 2013).

4.2 The Cell-vertex Grid Refinement

Refinements in ELBE are applied using the above-mentioned cell-vertex approach. Since moving overset grids are most often desired to be refined with respect to the stationary grid the topic is crucial for moving overset grids as well. In the following, the specific method by Filippova and Hänel (1998) will be discussed that is being used for refinements in ELBE and consequently also in this work. Implicitly a description of crucial details of the utilised overset method is given. Please note that two aligned grids (also referred to as patches) are assumed in this section. Requirements of arbitrarily orientated and moving patches will be described in subsequent sections.

4.2.1 Interpolation and Nested Time-stepping

Being an overset method, the concept of the cell-vertex refinement can generally be summed up as follows: Fluid properties of a coarse background grid are interpolated to the boundary nodes of a finer overset grid. Likewise fringe nodes of the coarse grid receive fluid properties from the fine grid. Also, inlying overlaid nodes of the coarse grid are usually deactivated in order to save memory and compute power, analogously to the hole-cutting in conventional overset grids. On top of that, however, certain additional procedural aspects arise due to the LBM that will be outlined in the following.

First of all refinement factor s is defined given as the ratio of the fine and coarse length scale with

$$s = \frac{\Delta x_c}{\Delta x_f} \quad . \tag{4.1}$$

Due to the acoustic scaling the same relation is given for the discrete time step and it follows that $\Delta t_f = \Delta t_c / s$. A refined subdomain consequently performs s nested time steps, meaning collisions and propagations, while the base grid performs only one (see figure 4.1). During the nested time-stepping there are no PDFs propagating to the boundary nodes

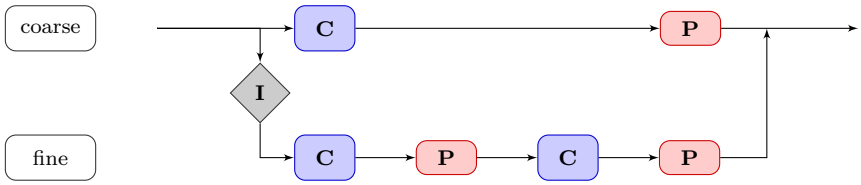


Figure 4.1: Procedural schematic of nested time-stepping on refined patches as illustrated by Gehrke (2015) for a refinement of $s = 2$. **C** denotes a collision step, **P** a propagation and **I** the interpolation.

of the fine patch. Missing distribution functions will consequently be found on the outer s layers of the fine patch. For a bidirectional coupling of two patches we consequently have to interpolate PDFs to the outer s node layers of the fine patch whilst only one layer of fringe nodes on the coarse patch requires PDFs from the fine patch. The latter is illustrated for the western boundary of a fine patch with $s = 2$ in figure 4.2.

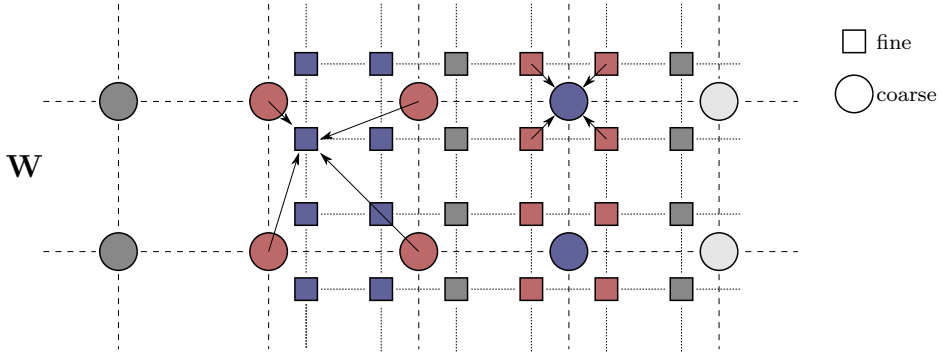


Figure 4.2: Illustrative schematic of donor and receptor nodes for the interpolation at the western boundary of a fine patch with $s = 2$. Donor nodes on each respective grid are marked in red, receptor nodes in blue. The light grey nodes on the coarse grid are deactivated.

In this work a direct interpolation of the post-collision PDFs is applied following a simple bi- or trilinear formulation for two- or threedimensional grids, respectively. The momentum is thus only recovered at first-order accuracy at the receptor node. However, note, that even a second-order interpolation of the momentum can be achieved by using only the direct neighbours as for the linear interpolation. The corresponding formulation of this so called complex interpolation is presented by Geier et al. (2009a). Instead of the PDFs they interpolate central moments. From the second-order moments which correlate with the stress tensor the gradient of the momentum at the neighbour nodes can be reconstructed which again yields a second-order formulation.

4.2.2 Scaling

A grid refinement is generally required to guarantee the same Reynolds number on all subgrids. Assuming a characteristic length scale L and velocity u both values can be scaled to the referring lattice units which yields

$$\mathrm{Re} = \frac{uL\,\Delta t}{\Delta x^2\,\nu} \tag{4.2}$$

where ν is the lattice velocity. Imposing $\mathrm{Re}_f = \mathrm{Re}_g$ and $\Delta x_f/\Delta t_f = \Delta x_c/\Delta t_c$ in accordance with the acoustic scaling we consequently obtain

$$\nu_f = s\,\nu_c \quad . \tag{4.3}$$

Referring to section 2.3.1 this again relates to a scaling of the relaxation time τ. With $\Delta t_f = \Delta t_c = 1$, τ_f can then be deduced from equation 2.18 which yields

$$\tau_f = 3\,s\,\nu_c + \frac{1}{2} \quad . \tag{4.4}$$

Apart from the error induced by the first-order interpolation, mass and momentum are conserved when propagating from grid to grid. In addition to that the scheme needs to preserve the continuity of the stress tensor. As shown in equation 2.22 to 2.24 the stress tensor correlates with the relaxation time. The scaling of τ therefore needs to be compensated for when it comes to the referring second-order moments. Mind, that in contrast to the formulation of the stress tensor given in section 2.3.2, Filippova and Hänel (1998) consult an equivalent notation based on the non-equilibrium PDFs, namely

$$S_{\alpha\beta} = -\left(1 - \frac{1}{2\tau}\right) \sum_{i=1}^{8} \boldsymbol{e}_{i\alpha}\boldsymbol{e}_{i\beta} f_i^{neq} \tag{4.5}$$

given here for the D2Q9 lattice. Its implementation is therefore, in a way, easier since no additional transformation into the moment space is required. By equating the stress tensor of the fine and coarse grid as given in equation 4.5 we obtain

$$\left(1 - \frac{1}{2\tau_f}\right) f_{i,f}^{neq} = \left(1 - \frac{1}{2\tau_c}\right) f_{i,c}^{neq} \tag{4.6}$$

which again allows us to deduce the required scaling factor $r_{c\to f}$ as

$$r_{c\to f} = \frac{\nu_c\,(6\nu_f + 1)}{\nu_f\,(6\nu_c + 1)} \quad . \tag{4.7}$$

The scaling factor for the opposite direction $r_{f\to c}$ is consequently given by the reciprocal value. Given the interpolated post-collision PDFs from the coarse grid $\hat{f}_{i,c}$ the final distribution functions passed to a fine node are given by

$$f_{i,f} = \hat{f}_{i,c}^{eq} + r_{c\to f}\left(\hat{f}_{i,c} - \hat{f}_{i,c}^{eq}\right) = \hat{f}_{i,c}^{eq} + r_{c\to f}\,\hat{f}_{i,c}^{neq} \quad . \tag{4.8}$$

For a receiving coarse node it follows straightforwardly that

$$f_{i,c} = \hat{f}^{eq}_{i,f} + r_{f\to c}\left(\hat{f}_{i,f} - \hat{f}^{eq}_{i,f}\right) = \hat{f}^{eq}_{i,f} + r_{f\to c}\,\hat{f}^{neq}_{i,f} \quad . \tag{4.9}$$

For the sake of completeness it should be mentioned that there is no limiting factor for multiple nested refinements. Assuming a second refined grid with a refinement factor s_2 relative to the first refinement level one obtains

$$\Delta x_2 = \frac{\Delta x_c}{s_1\, s_2} \quad , \tag{4.10}$$

where s_1 denotes the refinement factor of the first level. Moreover, for each individual refinement level the maximum for s is only determined by the stability criterion for the relaxation time mentioned in section 2.3.2. Usually the maximum for s is about 50. Most commonly, however, refinements of two to four are applied. For higher refinements one usually applies multiple overset grids as mentioned above.

4.3 The Description of Viscous Fluids in Moving Reference Frames

Further to the methodological fundamentals of overset grids in a LB framework, namely interpolation and scaling, arbitrary motions of the overlaying grid need to be accounted for. Most importantly this implies solving the governing equations in the local frame of reference of each grid, respectively. Even though we are eventually solving the LBE the objective is to reproduce the incompressible Navier-Stokes equations in inertial and especially non-inertial reference frames. Before discussing the non-inertial LBE, for the sake of completeness and clarity, a brief description of the Navier-Stokes equations in moving reference frames is provided.

4.3.1 Definition of Reference Frames and Kinematic Properties

In order to facilitate subsequent illustrations of methods and implementations, initially, a clear definition of the variables and transformations used in the different reference frames is given. First, an inertial reference frame K is defined which refers to a stationary background grid. The non-inertial reference frame, moving with an overset grid is defined as K'. For two-dimensional cases the origins of both coordinate systems coincide with the south-west vertex of the referring grid and in three-dimensional cases with the bottom-south-west vertex. Furthermore we follow the common notation of technical mechanics that all vectors in K are given as plain symbols while all vector in K' are given with a prime. Figure 4.3 illustrates the latter for the two-dimensional case including the kinematic properties of the moving patch, namely its position vector $\boldsymbol{p}$, translational velocity and acceleration $\dot{\boldsymbol{p}}$ and $\ddot{\boldsymbol{p}}$, respectively, rotation ψ and rotational velocity $\dot{\psi}$. Moreover $\boldsymbol{c}'$ is defined as the position of the centre of rotation and $\boldsymbol{n}'$ as the position of a node. The definition in three dimensions is of course given accordingly with z pointing from the

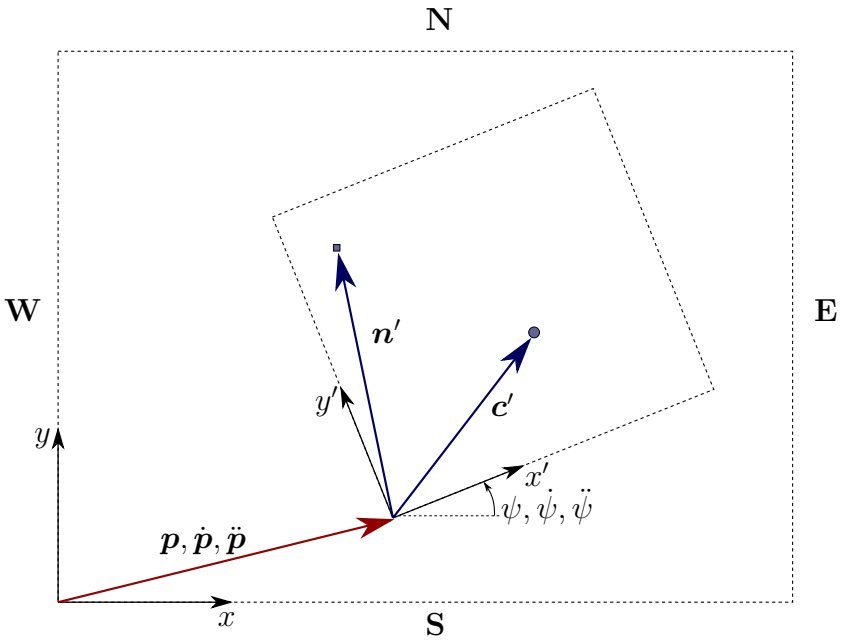


Figure 4.3: Definition of coordinate systems and kinematic properties in two dimensions.

bottom (**B**) to the top (**T**) of a patch. Here a rotation vector $\boldsymbol{r}$ with rotation angles about x, y and z, respectively, is defined as

$$\boldsymbol{r} = \begin{pmatrix} \varphi \\ \theta \\ \psi \end{pmatrix} \tag{4.11}$$

with the corresponding time derivatives being $\dot{\boldsymbol{r}}$ and $\ddot{\boldsymbol{r}}$.

The two-dimensional rotation transformation $\boldsymbol{R}_2$ rotating a vector $\boldsymbol{x}$ about z with ψ yields

$$\boldsymbol{x}_{rot} = \boldsymbol{R}_2(\psi)\,\boldsymbol{x} \quad \text{, with} \quad \boldsymbol{R}_2 = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \quad . \tag{4.12}$$

A rotation in the opposite direction can consequently be performed using $\boldsymbol{R}_2^{-1}$. In three dimensions we define $\boldsymbol{R}_3(\varphi, \theta, \psi)$ as follows:

$$\boldsymbol{R}_3(\varphi, \theta, \psi) = \boldsymbol{R}_z(\varphi)\,\boldsymbol{R}_y(\theta)\,\boldsymbol{R}_x(\psi) \quad , \tag{4.13}$$

where the matrices $\boldsymbol{R}_z(\psi)$, $\boldsymbol{R}_y(\theta)$ and $\boldsymbol{R}_x(\varphi)$ correspond to a rotation about the referring axis. Furthermore, note that rotated variables are also referred to with the subsript f for *fine* since the moving grid is usually refined with respect to the stationary *coarse* grid indicated by c. Moreover, fluid properties like $\boldsymbol{u}$ are generally subscripted with r for relative or a for absolute depending on the reference frame they refer to.

4.3.2 The Non-inertial Incompressible Navier-Stokes Equations

The general incompressible Navier-Stokes equations describe the conservation of momentum and mass of an infinitesimal volume of fluid. For inertial reference frames and without

any additional sink or source terms they are given by

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} \quad , \tag{4.14}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad . \tag{4.15}$$

Transforming the above-mentioned equations into a moving reference frame we start with the simple transformation of the position vector $\boldsymbol{x}$ which yields

$$\boldsymbol{x}' = \boldsymbol{R}^{-1}(\boldsymbol{x} - \boldsymbol{p}) \quad . \tag{4.16}$$

The time derivative of $\boldsymbol{x}'$ can then be obtained as

$$\boldsymbol{u}' = \boldsymbol{R}^{-1}(\boldsymbol{u} - \dot{\boldsymbol{p}}) - \dot{\boldsymbol{r}} \times \boldsymbol{x}' \quad , \tag{4.17}$$

where $\boldsymbol{u}'$ corresponds to the relative velocity in K, meaning $\boldsymbol{u}'_r$. Note that it is hereby assumed that the centre of rotation $\boldsymbol{c}'$ coincides with the origin of K'. Rearranging one obtains

$$\boldsymbol{u} = \dot{\boldsymbol{p}} + \boldsymbol{R}(\boldsymbol{u}' + \dot{\boldsymbol{r}} \times \boldsymbol{x}') \quad . \tag{4.18}$$

For the spatial derivatives $\partial/\partial x$, $\partial/\partial y$, $\partial/\partial z$, implicitly given as ∇, it can be deduced that

$$\nabla = \boldsymbol{R}\nabla' \quad \text{and} \tag{4.19}$$

$$\nabla^2 = \nabla'^2 \quad , \tag{4.20}$$

with the latter only holding for rotation transformations as in the given case. Consulting equations 4.18 to 4.20 one can substitute all absolute variables given in the NS equations. For the time derivative of the substituted $\boldsymbol{u}$ we obtain

$$\frac{\partial}{\partial t}\left(\dot{\boldsymbol{p}} + \boldsymbol{R}(\boldsymbol{u}' + \dot{\boldsymbol{r}} \times \boldsymbol{x}')\right) = \ddot{\boldsymbol{p}} + \ddot{\boldsymbol{r}} \times \boldsymbol{x}' + \dot{\boldsymbol{r}} \times (\dot{\boldsymbol{r}} \times \boldsymbol{x}') + 2\,\dot{\boldsymbol{r}} \times \boldsymbol{u}' + \frac{\partial \boldsymbol{u}'}{\partial t} \quad . \tag{4.21}$$

Here, the first four terms on the right hand side refer to the inertial forces due to the motion of the reference frame, namely the acceleration of the reference frame, the Euler-acceleration, the centrifugal acceleration and the Coriolis-acceleration, respectively. Simplifying and rearranging the convective, the diffusive and the pressure term as well as the continuity equation finally yields

$$\frac{\partial \boldsymbol{u}'}{\partial t} + (\boldsymbol{u}' \cdot \nabla')\boldsymbol{u}' = -\frac{1}{\rho}\nabla' p + \nu \nabla'^2 \boldsymbol{u}' + \boldsymbol{I}, \tag{4.22}$$

$$\boldsymbol{I} = -\ddot{\boldsymbol{p}} - \ddot{\boldsymbol{r}} \times \boldsymbol{x}' - \dot{\boldsymbol{r}} \times (\dot{\boldsymbol{r}} \times \boldsymbol{x}') - 2\,\dot{\boldsymbol{r}} \times \boldsymbol{u}' \quad , \tag{4.23}$$

$$\nabla' \cdot \boldsymbol{u}' = 0 \quad , \tag{4.24}$$

the Navier-Stokes equations in moving reference frames. Note that a very detailed description of all substitutions and rearrangements can be found in Li et al. (2002). Most

importantly, however, it can be summed up that the NS denoted in the arbitrarily orientated and moving system K' only differ from the inertial counterpart by the additional inertial forcing terms.

The consistency of the LBM with respect to the macroscopic NS equations can generally be shown using a Chapman-Enskog expansion or asymptotic analysis, see Krafczyk (2001) and Junk and Yang (2009a), respectively. As given above, solving the NS in a moving reference frame K' simply refers to solving in a different coordinate system while adding an extra forcing term on the right-hand side. Assuming the consistency of the applied forcing term in the LBM we can consequently conclude that solving the LBE in a moving reference frame is again consistent with the non-inertial NS equations.

4.4 The Lattice Boltzmann Method in Moving Reference Frames

In contrast to the NS equations, the LBE obviously necessitates the transformation of distribution functions to the reference frame of the receiving grid. This again involves two main problems that need to be addressed. First, the transformation has to map from one discrete to another discrete and not continuous space and, secondly, the advection correction at the grid interface involving the macroscopic patch velocity. Both will be outlined in the following including a brief introduction into moment spaces.

4.4.1 The Moment Space

Except for the BGK model, the collision step in the LBM is performed using velocity moments of the distribution functions. In that context (see section 2.3.2) the transformation of PDFs into the moment space was discussed as well as the macroscopic interpretation of the moments. As to be seen in the subsequent section, a rotation transformation of distribution functions given in a discrete velocity vector space can only be performed by means of a transformation into a moment space. For that reason the mathematical concept of moments shall be discussed in more detail and elaborate on the derivation of the transformation given in equation 2.20.

The general formulation of the n-th order moment of a continuous function f in one dimension is defined as

$$\mu_n = \int_{-\infty}^{\infty} (x - c)^n f(x) dx \quad . \tag{4.25}$$

More specifically the term *central* moments is used if c is set as the mean value of $f(x)$. On the other hand moments with $c = 0$ are usually referred to as *crude* or *raw* moments (Soong, 2004). The concept of moments is mostly found in statistical analysis. It shows, for instance, that the mean μ_1 of a single random variable is given by the first-order raw moment of $f(x)$. Or, we can find that the second-order central moment about the mean ($c = \mu_1$) is the variance of $f(x)$. Interestingly, the same formulation is applied in classical mechanics where the centre of gravity of a rigid body is obtained from the first-order raw moment of the mass and the moments of inertia from the second-order central moments of the mass. As to the latter, $c_{x,y,z}$ describes the distance to the axis through

the centre of mass. With $c_{x,y,z} \neq 0$ we consequently obtain a formulation better know as the Huygens-Steiner-theorem.

For the discrete velocity space of the LBM we can consequently define the raw moments for the D2Q9 and D3Q19 as

$$\mu_{\alpha\beta} = \sum_{i=0}^{8} \boldsymbol{e}_{ix}^{\alpha}\, \boldsymbol{e}_{iy}^{\beta}\, f_i \quad \text{and} \tag{4.26}$$

$$\mu_{\alpha\beta\gamma} = \sum_{i=0}^{18} \boldsymbol{e}_{ix}^{\alpha}\, \boldsymbol{e}_{iy}^{\beta}\, \boldsymbol{e}_{iz}^{\gamma}\, f_i \quad , \tag{4.27}$$

respectively. Note that central moments in the LBM are usually defined with c being the components of the macroscopic velocity $\boldsymbol{u}$, see for instance Geier et al. (2006, 2009b). Anyhow, as we shall only use raw moments in this work, central moments of the velocity space will not be explained in greater detail at this point.

Equations 4.26 and 4.27 reveal that the raw moments are actually given by n-th order polynomials of a two- or three-dimensional vector basis of 9 or 19 vectors, respectively. The linear transformation matrix given in equation 2.20 thus represents the component-wise evaluation of the underlying polynomial. This again refers to the moments being weighted sums of the values of the distribution functions with the weights being $\mu_{\alpha\beta}$ or $\mu_{\alpha\beta\gamma}$, respectively (Rubinstein and Luo, 2008). Depending on the choice of polynomials we can map the distribution functions onto different moment vector spaces. Equations 4.26 and 4.27 only refer to the most generic one only allowing monomials of the same order. The polynomials used for the D2Q9 in this work by Lallemand and Luo (2000) are given as

$$\rho_i = \boldsymbol{e}_{ix}^{0}\, \boldsymbol{e}_{iy}^{0}\,, \tag{4.28}$$

$$e_i = -4\, \boldsymbol{e}_{ix}^{0}\, \boldsymbol{e}_{iy}^{0} + 3\left(\boldsymbol{e}_{ix}^{2} + \boldsymbol{e}_{iy}^{2}\right)\,, \tag{4.29}$$

$$\epsilon_i = 4\boldsymbol{e}_{ix}^{0}\, \boldsymbol{e}_{iy}^{0} - \frac{21}{2}\left(\boldsymbol{e}_{ix}^{2} + \boldsymbol{e}_{iy}^{2}\right) + \frac{9}{2}\left(\boldsymbol{e}_{ix}^{2} + \boldsymbol{e}_{iy}^{2}\right)^2\,, \tag{4.30}$$

$$j_{x,i} = \boldsymbol{e}_{ix}^{1}\,, \tag{4.31}$$

$$q_{x,i} = \Big(- 5\boldsymbol{e}_{ix}^{0}\, \boldsymbol{e}_{iy}^{0} + 3\left(\boldsymbol{e}_{ix}^{2} + \boldsymbol{e}_{iy}^{2}\right)\Big)\, \boldsymbol{e}_{ix}^{1}\,, \tag{4.32}$$

$$j_{y,i} = \boldsymbol{e}_{iy}^{1}\,, \tag{4.33}$$

$$q_{y,i} = \Big(- 5\boldsymbol{e}_{ix}^{0}\, \boldsymbol{e}_{iy}^{0} + 3\left(\boldsymbol{e}_{ix}^{2} + \boldsymbol{e}_{iy}^{2}\right)\Big)\, \boldsymbol{e}_{iy}^{1}\,, \tag{4.34}$$

$$p_{xx,i} = \boldsymbol{e}_{ix}^{2} - \boldsymbol{e}_{iy}^{2}\,, \tag{4.35}$$

$$p_{xy,i} = \boldsymbol{e}_{ix}^{1}\, \boldsymbol{e}_{iy}^{1} \quad . \tag{4.36}$$

For the corresponding polynomials of the D3Q19 lattice it is again referred to Tölke et al. (2006). Note that even for the two lattices used in this work there are numerous

other moment spaces to be found in the literature. The common criterion for all moment spaces, however, is the mutually linear independence in particular, orthogonality (as well as weighted orthogonality, see e.g., Dellar, 2002; Geier et al., 2015), of all moments.

4.4.2 Transformation into Rotated Reference Frames

As a matter of fact the rotation of a local reference frame in a LB framework implies the rotation of the utilised discrete velocity set. In addition to the interpolation and scaling outlined in section 4.2 such a case necessitates the mapping of the PDFs onto a rotated vector space. It becomes obvious that a direct continuous rotation transformation of a discrete vector set is not possible. Mapping onto another discrete vector space, it is only straightforward to perform b discrete rotations of $2\pi/b$ about one axis, where b is the degree of rotational symmetry in that dimension (Chen and Orszag, 2011). Hence, the D2Q9 lattice is rotationally symmetric of degree eight. For the D3Q19 we obtain a rotational symmetry about the referring axis of $b_x = b_y = b_z = 4$. For any other rotation it is not possible to determine a unique transformation.

The problem, however, can be overcome by transforming the distribution functions into a moment space. Also it provides an interpretation of the validity of a continuous rotation with respect to the hydrodynamic variables, namely the order of isotropy where we shall follow the definition of Rubinstein and Luo (2008): Given a general moment of n-th order (referring to equation 4.25), the number of linearly independent components κ of that moment is equal to the number of linearly independent monomials of the corresponding polynomial. More precisely we obtain

$$\kappa_2 = (n+1) \quad \text{and} \quad \kappa_3 = \frac{1}{2}(n+2)(n+1) \tag{4.37}$$

in two and three spatial dimensions, respectively. Isotropy with respect to a certain discrete moment is thus given if the discrete moment provides the same amount of linearly independent components as its continuous counterpart. In this case the moment is defined as *complete* whereas missing components obviously lead to an *incomplete* moment. Accordingly the order of isotropy of a discrete velocity space refers to the order of the highest complete moment to be generated on the lattice.

As an example we consider the first order moments of the D2Q9 lattice representing the momentum in the x- and y-direction. These are obviously linearly independent, to be precise, orthogonal, and thus complete referring to equation 4.37. Correlating with a complete stress tensor it can furthermore be deduced that the second order moments of the D2Q9 are also complete. All higher order moments are incomplete resulting in a second-order isotropy which can also be determined for the D3Q19 lattice. Yet, second-order isotropy is sufficient to reproduce the incompressible NS equations since it captures their highest rank tensor, namely the stress tensor. All higher moments refer to so called *ghost* modes with respect to isothermal hydrodynamics (Dellar, 2003). Still, these may cause numerical instabilities if not treated correctly. In general, however, we expect degeneracies resulting from the rotation of anisotropic higher moments to be subordinate.

After a transformation into the particular moment spaces used in this work a rotation transformation can be applied. For the D2Q9 lattice an according formulation is presented by Gehrke (2015). The author derives the transformation based on the physical

interpretation of the moments. The first three moments, namely ρ, e and ϵ are found to be undirected and invariant to rotations. The first and third order moments refer the components of the momentum and energy flux, respectively and are consequently direction dependent. A rotation from K to K' is found to be straightforward using the rotation matrix $\boldsymbol{R}_2$ which yields

$$\rho\boldsymbol{u}(x',y') = \boldsymbol{R}_2^{-1}\,\rho\boldsymbol{u}(x,y) \quad , \tag{4.38}$$

$$\boldsymbol{q}(x',y') = \boldsymbol{R}_2^{-1}\,\boldsymbol{q}(x,y) \quad . \tag{4.39}$$

Note that the inverse of $\boldsymbol{R}_2$ is applied since a passive rotation about ψ is performed in order to transform into K'. For the transformation of the other second-order moments we have to consider the stress tensor $\boldsymbol{\sigma}$ which can be transformed according to

$$\boldsymbol{\sigma}(x',y') = \boldsymbol{R}_2^{-1}\,\boldsymbol{\sigma}(x,y)\,\boldsymbol{R}_2 \quad . \tag{4.40}$$

The first component $\sigma_{11}(x',y')$ can then written as

$$\sigma_{11}(x',y') = \left(\frac{1}{3}\rho c^2 + \frac{1}{2}p_{xx}(x',y') + \frac{1}{6}e - \rho_0 u_x^2(x',y')\right) \quad , \tag{4.41}$$

where we drop the term $-1-(1-\omega/2)$ (referring to equation 2.22) due to its occurrence on both sides of equation 4.40. Subsequently equation 4.41 can be rearranged to obtain

$$p_{xx}(x',y') = 2\left(\sigma_{11}(x',y') - \frac{1}{3}\rho c^2 - \frac{1}{6}e + \rho_0 u_x^2(x',y')\right) \quad , \tag{4.42}$$

where $\sigma_{11}(x',y')$ is inserted from equation 4.40 and $u_x(x',y')$ from equation 4.38. The remaining second-order moment p_{xy} can be determined accordingly. The entire rotation can then be summarised as the transformation matrix $\boldsymbol{T}_{c\to f}$ with

$$\boldsymbol{T}_{c\to f} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \cos\psi & 0 & \sin\psi & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \cos\psi & 0 & \sin\psi & 0 & 0 & 0 \\
0 & 0 & -\sin\psi & 0 & \cos\psi & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cos 2\psi & 2\sin 2\psi \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2}\sin 2\psi & \cos 2\psi
\end{pmatrix} \quad . \tag{4.43}$$

For the complete transformation of an interpolated and scaled fine distribution function $\boldsymbol{f}_f(x,y)$ it follows

$$\boldsymbol{f}_f(x',y') = \boldsymbol{M}^{-1}\,\boldsymbol{T}_{c\to f}\,\boldsymbol{M}\,\boldsymbol{f}_f(x,y) \quad . \tag{4.44}$$

The reverse transformation to obtain $\boldsymbol{f}_c(x,y)$ simply applies the inverse of $\boldsymbol{T}_{c\to f}$ which yields

$$\boldsymbol{f}_c(x,y) = \boldsymbol{M}^{-1}\,\boldsymbol{T}_{c\to f}^{-1}\,\boldsymbol{M}\,\boldsymbol{f}_c(x',y') \quad . \tag{4.45}$$

As outlined in section 2.3.2 the D3Q19 moment space comprises moments that do not have a straightforward interpretation in the context of the incomplete NS equations. A transformation can thus not be derived by means of the physical interpretation. Hence, we will follow a purely mathematical approach. Irrespective of the polynomial basis of a moment space we can choose an arbitrary set of coordinates for the integration of the underlying function, in this case the distribution function $f(\boldsymbol{x}, \boldsymbol{\xi})$. Starting from an explanatory simple raw moment of two variables with

$$\mu_{\alpha\beta} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^{\alpha} y^{\beta} f(x, y) dx dy \tag{4.46}$$

we can perform a change of variables and integrate over $dx'dy'$ instead. For now we assume a general transformation matrix $\boldsymbol{A}$ with the corresponding coordinate transformation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} x \\ y \end{pmatrix} \quad . \tag{4.47}$$

The change of variables yields an extra term in the integral, namely the determinant of the Jacobian of the transformation. It follows

$$\mu'_{\alpha\beta} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x'^{\alpha} y'^{\beta} f(x', y') \, \det \boldsymbol{J}(\boldsymbol{A}) \, dx' dy' \quad . \tag{4.48}$$

For this particular case we map $\boldsymbol{x}$ onto $\boldsymbol{x}'$. The transformation is therefore given by $\boldsymbol{A} = \boldsymbol{R}^{-1}$. As for all translation and rotation transformations we obtain $\det \boldsymbol{J}(\boldsymbol{A}) = 1$ which again simplifies the above-mentioned expression. Note, that an illustrative application of this procedure can be found in the field of image recognition. Integrating over a set of rotated basis vectors can identify a rotated image by means of rotation invariant moments, see for instance, Flusser et al. (2009). Here, the formulation will be used to derive a transformation that directly maps onto a rotated moment space.

As shown in equation 4.28 to 4.36 we use polynomials of the discrete velocity vectors $\boldsymbol{e}_i$ to derive the transformation into the moment space. Hence, it is directly integrated over a rotated basis by first applying the rotation transformation to the underlying velocity vectors with

$$\boldsymbol{e}'_i = \boldsymbol{R}^{-1} \, \boldsymbol{e}_i \quad . \tag{4.49}$$

and subsequently substituting $\boldsymbol{e}_i$ with $\boldsymbol{e}'_i$ in equation 4.28 to 4.36. Eventually one obtains $\tilde{\boldsymbol{M}}$ providing the transformation of the distribution functions into a moment space with respect to a set of rotated discrete velocity vectors. Referring to the above-mentioned procedure by Gehrke (2015) it can be shown that

$$\tilde{\boldsymbol{M}} = \boldsymbol{T}_{c \to f} \, \boldsymbol{M} \quad . \tag{4.50}$$

The complete transformation into the rotated reference frame thus becomes

$$\boldsymbol{f}_f(x', y') = \boldsymbol{M}^{-1} \, \tilde{\boldsymbol{M}} \, \boldsymbol{f}_f(x, y) \tag{4.51}$$

and accordingly

$$\boldsymbol{f}_c(x, y) = \boldsymbol{M}^{-1} \, \tilde{\boldsymbol{M}}^{-1} \, \boldsymbol{f}_c(x', y') \tag{4.52}$$

for the reverse transformation. It becomes obvious that the stated method does not require any physical interpretation of the moments. The above-mentioned problem of the D3Q19 lattice can thus be overcome starting from the referring transformation of the three-dimensional velocity vectors with $\boldsymbol{e}_i' = \boldsymbol{R}_3^{-1}\,\boldsymbol{e}_i$. Moreover it is straightforward to apply the procedure to any other lattice like for instance the D3Q27.

4.4.3 Advection Correction

The transformation at the grid interface into a possibly moving reference frame relates to a pseudo change of momentum that needs to be incorporated. When transforming from the coarse to the fine grid (inertial to moving reference frame) the interpolated PDFs $\boldsymbol{f}_c$ represent the absolute momentum on the coarse gird, hence absolute velocity $\boldsymbol{u}_a$, which is given by

$$\boldsymbol{u}_a(\boldsymbol{x}) = \dot{\boldsymbol{p}} + \boldsymbol{R}\left(\boldsymbol{u}_r'(\boldsymbol{x}') + \dot{\boldsymbol{r}} \times (\boldsymbol{x}' - \boldsymbol{c}')\right) \quad , \tag{4.53}$$

where $\boldsymbol{u}_r'(\boldsymbol{x}')$ denotes the relative fluid velocity in the local reference frame as derived in section 4.3.2. In other words, we have to obtain a set of distribution functions referring to $\boldsymbol{u}_r'$ from $\boldsymbol{f}_c$ and, for the reverse transformation a set of PDFs referring to $\boldsymbol{u}_a$ from $\boldsymbol{f}_f'$. Here we will calculate $\boldsymbol{f}_f$ referring to $\boldsymbol{u}_r$ (the relative velocity given in absolute coordinates K, not K') and apply the coordinate transformation derived in section 4.4.2 subsequently. Again, the same applies for the reverse transformation. The relative velocity in K is thus given by

$$\boldsymbol{u}_r(\boldsymbol{x}) = \boldsymbol{u}_a(\boldsymbol{x}) - \left(\dot{\boldsymbol{p}} + \dot{\boldsymbol{r}} \times (\boldsymbol{x} - \boldsymbol{c})\right) = \boldsymbol{u}_a(\boldsymbol{x}) - \boldsymbol{u}_p(\boldsymbol{x}) \quad . \tag{4.54}$$

The second term $\boldsymbol{u}_p(\boldsymbol{x})$ now denotes the absolute velocity of a point $\boldsymbol{x}$ on the moving patch. Moreover, it can be shown that $\boldsymbol{u}_p$ is inviscid, meaning irrotational, following

$$\nabla \times \boldsymbol{u}_p = 0 \quad . \tag{4.55}$$

The latter appears to be straightforward since the patch performs a rigid body motion and is consequently not sheared.

With $\boldsymbol{u}_p$ being inviscid we only have to correct the equilibrium part of the interpolated PDFs. This implies that the final set of corrected PDFs $\boldsymbol{f}_{c,r}$ must fulfill

$$\boldsymbol{f}_{c,r}^{eq} \stackrel{!}{=} \boldsymbol{f}_c^{eq}(\boldsymbol{u}_r, \rho) \quad \text{and} \tag{4.56}$$

$$\boldsymbol{f}_{c,r}^{neq} \stackrel{!}{=} \boldsymbol{f}_c^{neq} \quad . \tag{4.57}$$

In what follows it becomes crucial to bear in mind that the Maxwellian equilibrium and its Taylor expansion, accordingly, are non-linear with respect to the macroscopic velocity $\boldsymbol{u}$ (see, equations 2.13 and 2.14, respectively). Directly subtracting $\boldsymbol{f}_c^{eq}(\boldsymbol{u}_p, \rho)$ from $\boldsymbol{f}_c$ consequently violates equation 4.56 since it introduces cross-coupling terms of the relative and absolute velocity. The latter becomes obvious from

$$\begin{aligned} f_c^{eq}(\boldsymbol{u}_a - \boldsymbol{u}_p, \rho) &= \zeta\, \exp - \frac{(\boldsymbol{\xi} - (\boldsymbol{u}_a - \boldsymbol{u}_p))^2}{2c_s^2} \\ &= f_c^{eq}(\boldsymbol{u}_a, \rho) + \zeta\, \exp - \frac{(\boldsymbol{u}_p \cdot \boldsymbol{u}_p - 2\,\boldsymbol{u}_a \cdot \boldsymbol{u}_p + 2\,\boldsymbol{\xi} \cdot \boldsymbol{u}_p)}{2c_s^2} \end{aligned} \tag{4.58}$$

$$\neq f_c^{eq}(\boldsymbol{u}_a, \rho) - f_c^{eq}(\boldsymbol{u}_p, \rho) = f_c^{eq}(\boldsymbol{u}_r, \rho) \tag{4.59}$$

$$\Leftrightarrow f_c^{eq}(\boldsymbol{u}_p, \rho) = f_c^{eq}(\boldsymbol{u}_a, \rho) - f_c^{eq}(\boldsymbol{u}_r, \rho) \quad , \tag{4.60}$$

with $\zeta = \rho/(2\pi c_s^2)^{D/2}$. Following equation 4.60 the absolute velocity can be calculated as

$$\boldsymbol{u}_a = \boldsymbol{u}_a(\boldsymbol{f}_c) \tag{4.61}$$

as well as the final relative velocity $\boldsymbol{u}_r$ using equation 4.54 with $\boldsymbol{u}_a(\boldsymbol{f}_c)$. After all we can reconstruct the according equilibria $\boldsymbol{f}_a^{eq}(\boldsymbol{u}_a, \rho)$ and $\boldsymbol{f}_r^{eq}(\boldsymbol{u}_r, \rho)$ using the simplified Maxwellian equilibrium distribution and perform the final advection correction, with

$$\boldsymbol{f}_{c,r} = \boldsymbol{f}_c - \left(\boldsymbol{f}_a^{eq}(\boldsymbol{u}_a, \rho) - \boldsymbol{f}_r^{eq}(\boldsymbol{u}_r, \rho)\right) \quad , \tag{4.62}$$

fulfilling the continuity of the stress tensor implied in equation 4.57 as well as the pseudo change of momentum due to the patch motion, referring to equation 4.56.

For the reverse transformation the momentum implied in the patch motion has to be added to the interpolated fine PDFs $\boldsymbol{f}'_f$. This gives us

$$\boldsymbol{f}'_{f,a} = \boldsymbol{f}'_f + \left(\boldsymbol{f}'^{eq}_a(\boldsymbol{u}'_a, \rho) - \boldsymbol{f}'^{eq}_r(\boldsymbol{u}'_r, \rho)\right) \quad , \tag{4.63}$$

where $\boldsymbol{f}'_{f,a}$ denotes the distribution functions referring to the absolute momentum given in K'. In this case we obviously reconstruct $\boldsymbol{u}'_r$ from $\boldsymbol{f}'_f$ and obtain $\boldsymbol{u}'_a$ by adding the patch velocity $\boldsymbol{u}'_p(\boldsymbol{x}')$.

4.4.4 Application of Inertial Forces

The derivation of the non-inertial NS equations reveals that additional inertial forces occur when solving in a local moving reference frame. Within the LBE these can be incorporated using the formulation provided in section 2.5. Equation 2.33 then reads

$$F_i = 3w_i\,\rho\,\boldsymbol{e}_i \cdot \boldsymbol{I} \quad , \tag{4.64}$$

with $\boldsymbol{I}$ being the sum of inertial forces given in equation 4.23. The evaluation of $\boldsymbol{I}$ is thus relatively straightforward and can be performed in every discrete time step prior to the collision step. Apart from the translational and rotational velocities and accelerations of the patch we only require the relative fluid velocity at $\boldsymbol{x}'$ for the evaluation of the Coriolis force. Due to the current computational procedure it is deduced from the pre-collision PDFs which yields

$$\boldsymbol{I}_c = 2\,\dot{\boldsymbol{r}} \times \boldsymbol{u}'_r\left(\boldsymbol{f}'(\boldsymbol{x}')\right) \tag{4.65}$$

for the Coriolis term. As we have seen in section 4.3.2 the velocity dependence of the Coriolis term states the only dependence of the inertial forcing to a fluid property. The use of the pre-collision velocity is thus somewhat arbitrary and only reasoned by the call of the so called `calcForcing` visitor (see chapter 5) prior to the collision step. A further analysis of this term in conjunction with the applied LB forcing formulation will have to clarify the validity of this choice.

Chapter 5

Implementation

Based on the preceding mathematical description of moving overset grids this chapter will provide an outline of the implementation within the GPU-accelerated LB solver ELBE. After an introductory summary of the state of the art in ELBE a procedural overview will be given of the newly developed overset method within the existing framework. Moreover, it will be elaborated in greater detail on specific algorithms, in particular, the identification of interface nodes.

5.1 Initial Development Situation in ELBE

The LB solver ELBE is a general purpose CFD tool for the simulation of viscous flows based on the LBM. Its main applications are technical and maritime single-phase and free-surface flows. It comprises functionalities such as bi-directional fluid-structure coupling, online visualisation as well as a LES turbulence model. The code is developed for the massive parallelization on graphic processing units using the Nvidia-proprietary parallel computing platform CUDA as an extension to C++.

5.1.1 Moving Bodies in ELBE

The focus on maritime applications makes the simulation of moving bodies and their interaction with single-phase or free-surface flows inevitable. Besides prescribed uncoupled rigid body motions ELBE allows for a bidirectional coupling using an explicit fluid structure interaction (FSI) approach. Based on the fluid loads the rigid body motion is computed using the open-source Open Dynamics Engine, see Mierke (2015). The forcing on the fluid is generally incorporated using LIBB or QIBB boundary conditions. Two exemplary simulation results are given in figure 5.1.

With the use of bounce-back schemes the motion of a body is considered by node updating as described in section 3.3.2. The problem of spurious pressure oscillations due to the reinitialisation of fluid nodes can thus also be found in ELBE and is addressed throughout this work.

5.1.2 Program Structures for Moving Overset Grids

Grid refinements as currently implemented in ELBE are based on the method by Filippova and Hänel (1998). As outlined in section 4.2 this method, by definition, refers to overset grids and thus provides certain class and member definitions that we shall follow in this

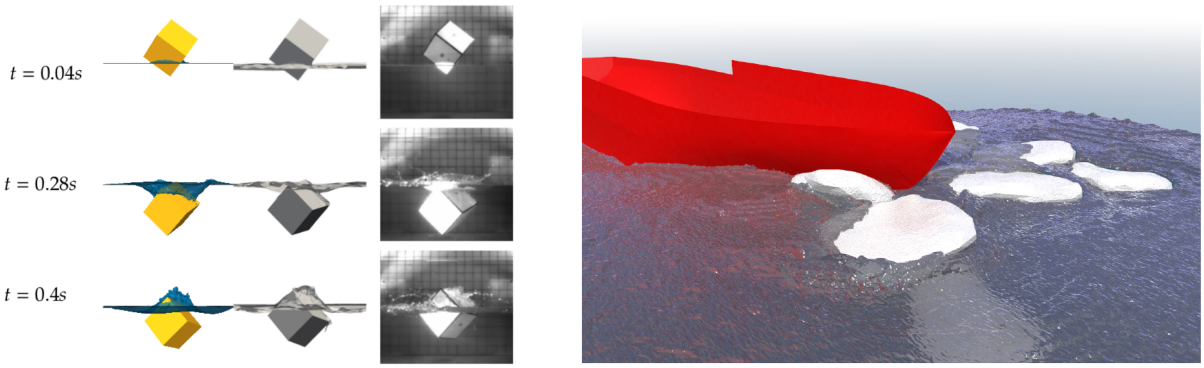


Figure 5.1: Exemplary FSI simulation results of ELBE: (a) Comparison of numerical results from ELBE (left) and the FV code FresCo$^+$ (middle) with experimental results (right) of a freely falling, ditching cube. Source: Janßen et al. (2015). (b) Ship-ice interaction of the ice-going ship Polarstern. Source: Mierke et al. (2015).

work. For the sake of clarity within following discussions in this chapter we shall give a brief description of the underlying terms and concepts: The most important ones are the `patch` and `connector` class and their derived classes for the corresponding solvers like `elbeSP3D` (single-phase 3D) or `elbeFS3D` (free-surface 3D).

Here, a patch generally refers to a subdomain, meaning an equidistant Cartesian grid. Apart from members containing the PDFs or macroscopic variables a patch is characterised by its refinement level s with $s = 1$ referring to the base patch of a simulation set-up. By default the origin of the base patch marks the origin of the global coordinate system. The `origin` objects of the refined patches consequently denote their corresponding global position.

A connector couples two patches of different refinement levels. In other words it performs the interpolation and scaling procedures outlined in section 4.2. In ELBE the overall overset coupling is separated into a coarse-to-fine and a fine-to-coarse procedure performed by a `CFConnector` and a `FCConnector` class, respectively.

Apparently the introduction of patch motions to an established code requires the addition of numerous parameters, or to be precise, members to existing classes. Moreover, existing utility functions need to be adapted in order to include rotations or patch velocities. Mind, that without the introduced patch motions the coordinate systems of two patches are always aligned and only differ by the scale of the underlying lattice units. In order to actually change the orientation of a patch during a simulation we define the visitor-type class `movePatch` that again accesses patch members via corresponding public getter and setter functions. The same concept is applied for the calculation and application of the inertial forces introducing a `calcForcing` visitor.

5.2 Procedural Overview

In the following an outline will be given of the overall procedure of solving the LBM on a moving overset grid. We hereby assume the exemplary case of one overset patch of refinement level s moving within the base patch. Referring to section 4.2 the overset patch consequently performs s time steps of length $\Delta t'$ within one nested time loop. A schematic of a nested time loop is given in figure 5.2 and shall illustrate to subsequent explanation.

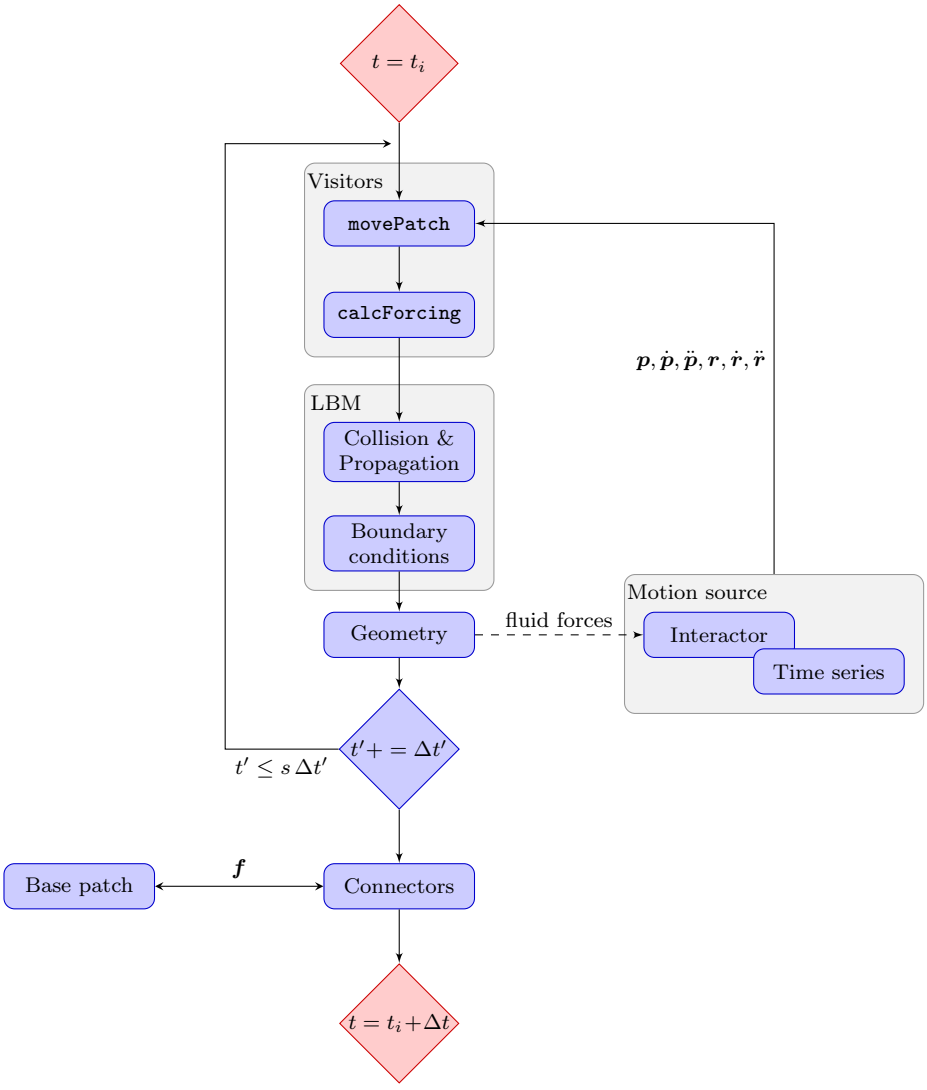


Figure 5.2: Procedural flow-chart of a nested time loop of a moving overset patch

Within the overall simulation we assume the time $t = t_i$. Apart from other visitor objects (like probes etc.) the `movePatch` and `calcForcing` visitors are invited by the patch. The `movePatch` visitor retrieves all kinematic properties of the patch from either a so called interactor or a subsidiary time series. The interactor is an interface class coupling the LBM solver with a rigid body solver or other steering plug-ins. Hence,

it provides kinematic properties of a geometry object based on the fluid loads on the geometry of the preceding time step or simply following a time dependent function (for further details, see Mierke, 2015). Since the patch is usually said to follow the geometry object's motion the kinematic properties coincide. Note, however, that the rotation of the patch is defined about its origin which may differ from the centre of rotation of the object. An according transformation is therefore performed by the `movePatch` visitor. Finally, it calls a corresponding setter function of the patch and forwards the updated kinematic properties.

Subsequently the `calcForcing` visitor computes the node-wise inertial forcing based on the patch kinematics. In order to accelerate the procedure it is executed as a CUDA kernel with one thread per node, respectively.

The subsequent step refers to the standard collision and propagation step of the LBM including the application of boundary conditions. In this case the latter only refers to no-slip (bounce-back) boundary conditions of included geometries. Boundary conditions at the outer subdomain boundary are rather unusual and would replace the above-mentioned connectors. As we have seen in section 4.2 the PDFs at the outer node layers usually stem from the overlaid grid. Hybrid procedures though are technically conceivable but not considered in this work.

In the case of a bidirectional FSI case, it follows the computation of fluid loads on the geometry object. Most commonly, the forces are calculated using the momentum exchange method (MEM) referring to Nguyen and Ladd (2005). For the following time step these are now being provided as members of the geometry object.

Terminating the nested time loop the connectors are called refilling the PDFs at the outer node layers of the overset patch and providing the reverse coupling to the base patch. Note that all transformations of the interpolated PDFs as well as the advection correction are included in the connector objects. These shall be further elaborated on in the subsequent sections. Moreover, it should be mentioned that, at the time of writing, hole-cutting is not considered. The LBE is consequently still solved on nodes overlaid by the finer patch. For the simulation itself the fluid properties obtained here are irrelevant. It remains unclear whether the implementation of hole-cutting is actually beneficial for the overall efficiency. After all, the effort for the hole-cutting itself must be less than solving the overlaid nodes. This again may not be the case due to the massive parallelization in ELBE. Hence, only the conservation of memory is guaranteed. On top of that hole-cutting again implies the reinitialisation of fluid nodes whose prevention is one of the main motivations for moving overset grids in the LBM.

5.3 Identification of Interface Nodes

Finding the correct nodes for the interpolation of fluid properties to nodes of the grid interface is one of the main efficiency challenges for overset methods in general. Concerning this matter the equidistant Cartesian grids utilised in ELBE and the computational molecule of linear interpolations combine the two least problematic cases. Still, for high-performance codes like ELBE computational efficiency is particularly crucial. An optimal exploitation of the computational resources of GPUs by means of massive parallelization should thus be intended. In the following we will discuss the algorithms developed for

this purpose. Not only because of their historical separation within the code, but mainly due to their noticeably different nature the coarse-to-fine and the fine-to-coarse algorithm will be treated separately.

5.3.1 The Coarse-to-fine Interface

Using an equidistant Cartesian grid provides us with a vital advantage for the localisation of nodes in general. Given an arbitrary absolute coordinate in SI units we only need to scale it with the corresponding normalisation factor N_x in order to obtain the coordinates in lattice units. Rounding the real-valued coordinates to the upper or lower integers then again yields the referring node number of the lattice. Here, $(0\,0)^T$ or $(0\,0\,0)^T$ refers to the first node of the two- or three-dimensional lattice, respectively, located at the lattice origin. Given an arbitrarily oriented overset grid as depicted in figure 5.3 a receptor node position $\boldsymbol{n}^{rec}_{c\to f,c}$ in coarse lattice units can be determined as

$$\boldsymbol{n}^{rec}_{c\to f,c} = N_{x,c}\left(\boldsymbol{p} + \frac{1}{N_{x,f}}\,\boldsymbol{R}(\boldsymbol{r})\,\boldsymbol{n}^{rec'}_{c\to f,f}\right) \quad , \tag{5.1}$$

with $N_{x,c}$ and $N_{x,f}$ denoting the spatial normalisation factors for the coarse and fine grid, respectively.

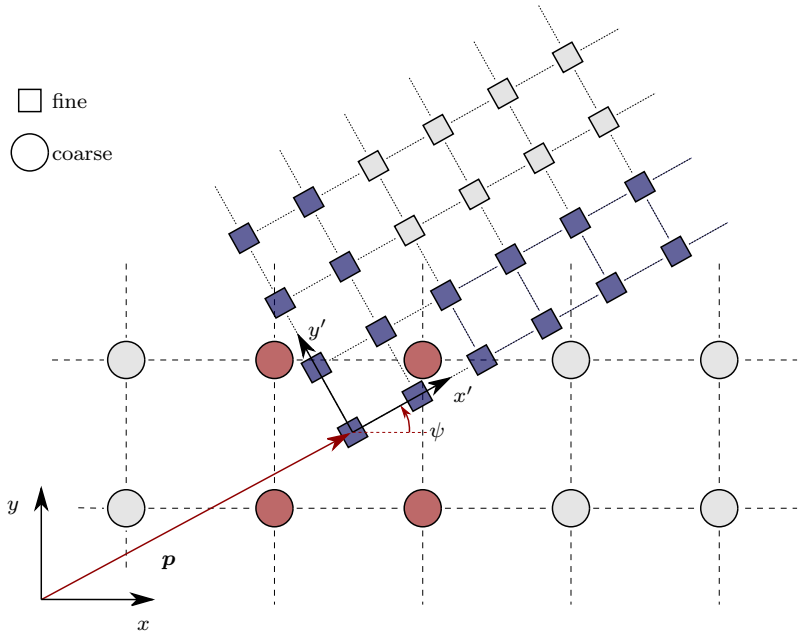


Figure 5.3: Illustrative schematic of coarse-to-fine interface nodes of a two-dimensional overset grid with $s = 2$. Shown is the south-west corner of the overset grid including its origin. Receptor nodes on the overset grid are marked in blue. Exemplary donor nodes on the background grid for the node at $(0\,0)'^T$ are marked in red.

Subsequently the components of $\boldsymbol{n}^{rec}_{c\to f,c}$ are rounded to the lower integers by type-casting yielding the south-west donor node on two-dimensional lattices and the bottom-south-west node on three-dimensional lattices, respectively. The other three or seven nodes for the

interpolation can then be accessed by adding corresponding integers to the memory index obtained for the SW or BSW node.

The entire procedure is thus very straightforward, especially since the receptor nodes on the fine grid are known a priori. Throughout an entire simulation they represent the outer s node layers on the overset grid and must therefore only be identified at the initialisation. The latter implies that the connector kernels always require the same amount of threads and only need to find the surrounding donor nodes following the above-mentioned procedure.

5.3.2 The Fine-to-coarse Interface

Identifying the fine-to-coarse interface is far more computationally demanding than for the other interface. That is because neither the receptor nor the donor nodes can be known a priori. We partially discussed the problem in section 3.2.1 as part of the hole-cutting procedure for overset grids in general. All in all, the fine-to-coarse interface underlies the following conditions:

- The receptor nodes on the coarse grid may not coincide with the donor nodes on this grid.
- The fringe of receptor nodes shall directly border the fringe of donor nodes in order to keep the overset area as small as possible.
- The fringe of receptor nodes must be closed with respect to all discrete velocity directions.

The identification of coarse receptor nodes thus obviously depends on the position of the coarse donor nodes which again depends on the position of the overset grid. Hence, the motion of the overset grid makes the fine-to-coarse interface time-dependent which not only refers to the location of its nodes but also to the number of nodes. Finding the interface is consequently performed at each discrete time step comprising several sequential kernel functions (as opposed to only one for the coarse-to-fine node identification) that will be discussed in the following.

A direct analytical description of the fringe of donor nodes, e.g. using techniques like Bresenham's line algorithm (Bresenham, 1965), is generally difficult when incorporating all of the above-mentioned criteria. It might seem appealing to simply follow a straight line, in this case parallel to the coarse-to-fine interface, along which one generally valid function identifies the desired nodes. It shows, however, that the choice of the start and end point of the line as well as the spatial discretisation for the evaluation of the function are again dependent on the position and rotation of the overset grid. For the sake of robustness we consequently apply a rather heuristic method. Nevertheless, the possibility of a robust analytical description can not be excluded.

The overall procedure can be separated into three main steps, namely identifying the inner donor nodes of the coarse-to-fine interface (which must in any case be excluded from this interface), flagging the fine-to-coarse receptor nodes and finally identifying the corresponding fine donor nodes. The two former steps are illustrated for an exemplary two-dimensional grid in figure 5.4.

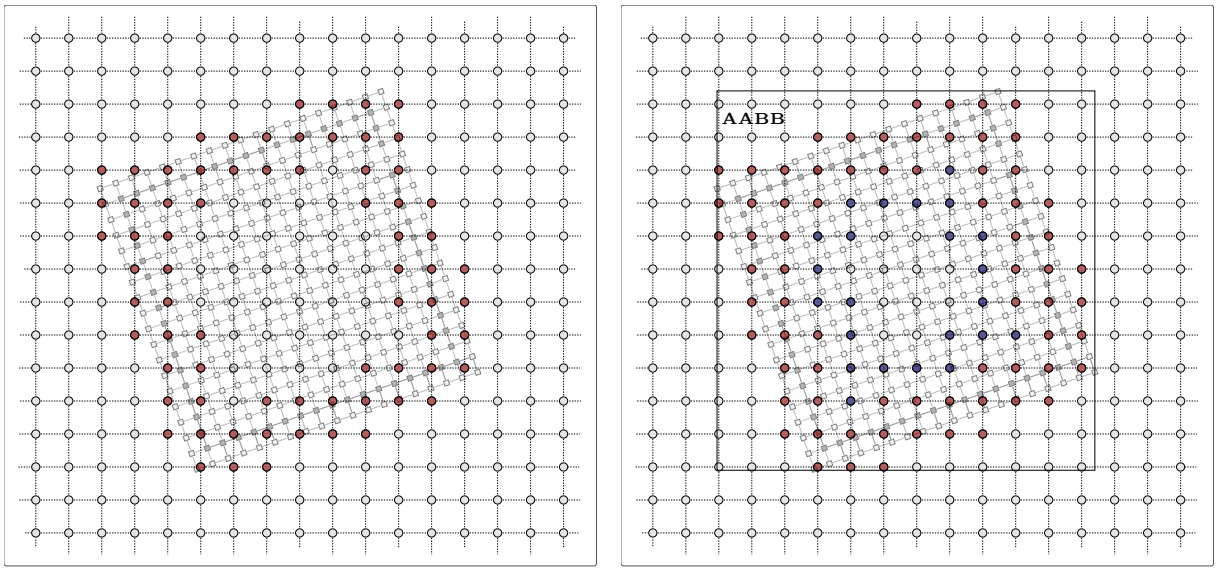


Figure 5.4: Illustrative schematic of the fine-to-coarse interface identification on a two-dimensional grid (overset refinement factor $s = 2$). (a) Starting from the inner coarse-to-fine receptor nodes (dark grey) the corresponding donor nodes are flagged (red). (b) A subsequent kernel flags the final receptor nodes (blue) based on the previous selection. The AABB of the rotated patch marks the search domain for the `DeviceSelect` kernel gathering the memory indices of the final interface nodes.

An initial kernel marks the donor nodes for the inner layer of fine nodes of the coarse-to-fine interface. The corresponding procedure is thus equivalent to the node identification outlined in the preceding section but with less threads needed.

The subsequent kernel marks the actual receptor nodes on the coarse grid. The starting point of the identification algorithm, again, is the inner layer of fine nodes of the coarse-to-fine interface. Depending on the edge or face of the patch, respectively, the kernel is provided with two/three offset vectors, a normal search and one/two tangential search vector. Hereby, the algorithm is designed to initialise the search for interface nodes always from the same relative point, namely the adjacent SW or BSW coarse node of the corresponding fine start node. The spatial dependence of the process is entirely incorporated in the offset and search vectors being arguments of the function. The process can thus be perfectly parallelized. In conjunction with figure 5.5 the details of the individual steps are outlined as follows:

1) Identification of adjacent SW or BSW node analogously to the procedure outlined in section 5.3.1.

2) Skip to the inner nodes of the coarse-to-fine interface (two nodes in 2D, four nodes in 3D) by adding the two offset vectors (in 2D) or linear combinations of the three offset vectors (in 3D)

3) Proceed in the normal search direction. If the node is not identified as a coarse-to-fine node it is a fine-to-coarse receptor node and marked accordingly. Note that this

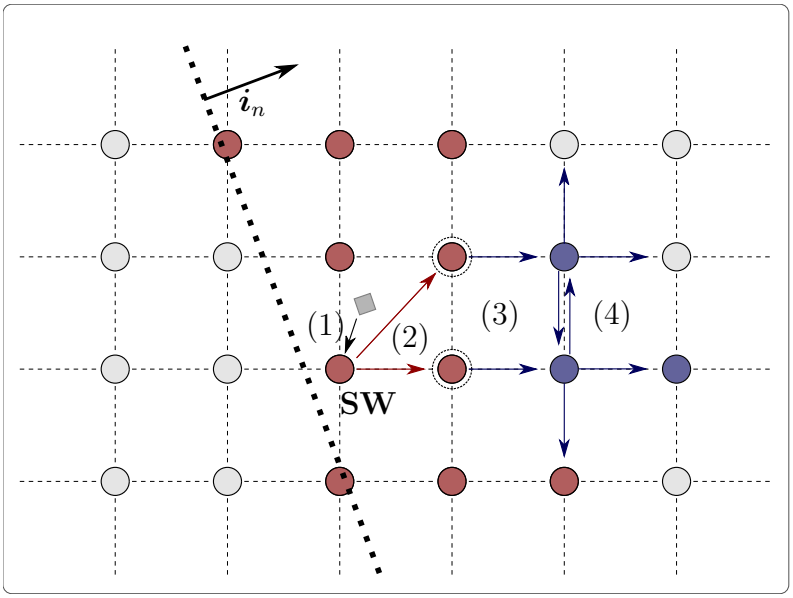


Figure 5.5: Illustration of the fine-to-coarse interface identification algorithm. The dotted line marks the West boundary of the exemplary overset patch with its normal vector $\boldsymbol{i}_n$. Coarse-to-fine nodes are marked in red with the nodes skipped to by the offset vectors being additionally circled. Positively identified interface nodes of the shown thread are marked in blue.

step generally always finds an interface node. Only if the thread is started in the corners of a patch an exception can occur.

4) Proceed further in the normal search direction and mark the node if one node in the tangential search directions is identified as a coarse-to-fine node. This additional condition in the tangential direction assures that nodes are identified that are necessary for a closed interface. See for instance the most easterly node flagged in the example in figure 5.5. A thread starting in the row south of the example thread would only mark the first node in the normal direction, leaving a gap in the interface in the discrete NE direction.

Also note that the determination of the offset and search directions follows simple geometric relations based on the normal vector of the interface edge or face $\boldsymbol{i}_n$. Here, the largest component of $\boldsymbol{i}_n$ is determined by means of magnitude as well as the corresponding direction, simply given by the sign of the component. The largest component consequently determines the closest discrete lattice direction where again only the non-diagonal directions are considered. For a two-dimensional lattice we consequently obtain a discrete interface normal $\boldsymbol{i}_n^d$ as

$$\boldsymbol{i}_n^d = \begin{cases} \begin{pmatrix} \operatorname{sgn}(\boldsymbol{i}_{n,x}) \cdot 1 \\ 0 \end{pmatrix} & , \boldsymbol{i}_{n,x} \geq \boldsymbol{i}_{n,y} \\ \begin{pmatrix} 0 \\ \operatorname{sgn}(\boldsymbol{i}_{n,y}) \cdot 1 \end{pmatrix} & , \boldsymbol{i}_{n,x} < i_{n,y} \end{cases} . \tag{5.2}$$

Using the resulting discrete interface normal direction we can determine the adjacent inner interface nodes by means of the offset vectors (encircled nodes in figure 5.5). These are not explicitly transferred to the kernels. Beforehand they are translated to corresponding node memory index offsets relative to the starting SW/BSW nodes. The data per vector is thus reduced from two or three real values, respectively to one integer. The same applies to the normal and tangential search vectors, where the normal search vector is equal to the discrete interface normal and the tangential search vector is the corresponding orthogonal.

With all interface nodes being flagged the actual interpolation, scaling and transformation of PDFs can be performed for the referring nodes. As a last preparatory step, however, the memory indices of the flagged nodes are assembled in a sorted array. Threads of the final kernel performing the aforementioned tasks can thus be assigned only to interface nodes without requiring another check of the flag of the node. And, most importantly, the kernel is not started with more threads than actually needed. For the preparatory gathering of memory indices the `DeviceSelect` function of the open-source library CUB (NVIDIA, 2015) is applied. In order to minimise the computational effort, the search domain for the `DeviceSelect` kernel is limited to the AABB of the overset patch as shown in figure 5.4b.

Chapter 6

Verification and Error Analysis

Following the discussion by **?** the verification of the developed method refers to answering whether we are '*solving the equations right*'. With respect to fluid dynamics this obviously refers to solving the NS equations. It shall thus be determined and discussed whether the transformations and interpolations etc., implemented as part of the overset grid method still recover the governing equations. In this case we specifically need to focus on the errors induced in addition to those inherently made by the applied LBM. For that purpose several test series are performed trying to isolate the errors of each individual part of the entire method.

6.1 Error Analysis in Spatially Constant Flows

Irrespective of the main motivation for overset grids, namely the simulation of moving bodies, first, it shall be determined how the method itself perturbs a simple flow without including a rigid body. The most simple example thereof would be a moving patch within a calm body of fluid. The problem hereby is that there is no straightforward way to put erroneous velocities or density fluctuations observed within the fluid into perspective. Mind, all reference quantities are zero. Moreover, in many test scenarios the errors induced accumulate in certain regions of the domain, e.g. when applying a simple patch rotation about one fixed point. Here, the instantaneously induced errors can hardly be isolated from the errors induced by preceding time steps. Ultimately, these only propagate with velocities of the order of their own magnitude which is, ideally, small compared to the patch motion. For the verification of the method we shall therefore place the overset patch within a domain with constant velocity. For each test series the velocity or density profile of a reference simulation without overset grid is consulted in order to give measures of the relative error of the moving overset grid.

6.1.1 Case Set-up and Method of Analysis

For all cases of this study a cubic overset patch of edge length $L_f = 1/10\, L_c$ is placed in the middle of a channel of width L_c. The refinement of the patch is set as $s = 2$. Furthermore, we introduce the patch velocity ratio Φ_p with

$$\Phi_p = \frac{\dot{\boldsymbol{p}}_x}{u_{ref}} \quad . \tag{6.1}$$

The time step size of the applied LBM is generally determined using the reference velocity u_{ref}. Φ_p thus provides a measure of the magnitude of the relative velocity on the moving patch with respect to the time step which is chosen in order to provide a stable LB simulations. With $\Phi_p = 0$ the patch is obviously not moving and sees the same velocity as the background patch, with $\Phi_p = 1$ the relative velocity on the patch is zero and for $\Phi_p > 2$ the relative velocity on the patch is larger than the reference velocity which is expected to lead to stability problems. Moreover, we define the Reynolds number of this problem using the overset patch length L_f ($\mathrm{Re} = u_{ref}\, L_f/\nu$). A summary of all test case properties used throughout the entire verification is given in table 5.1 with an additional illustration being provided in figure 5.6.

Table 6.1: Test case properties of the verification procedure

Overset patch length	L_f	$0.1\,\mathrm{m}$
Refinement factor	s	2
Background patch width	L_c	$1.0\,\mathrm{m}$
Reynolds number	Re	1000
Viscosity	ν	$10^{-5}\,\mathrm{m^2\,s^{-1}}$

All parameters not given here vary between the different test cases and will be explicitly given for the referring case. Also note that for the background grid all boundaries are provided with velocity boundary conditions with u_{ref}, except for the outlet where an extrapolation boundary condition is applied.

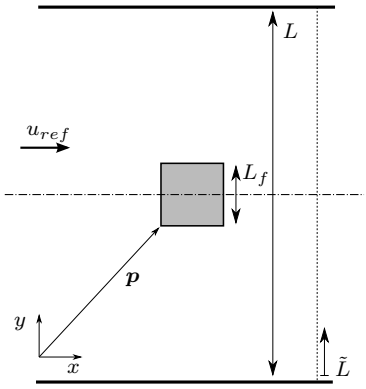


Figure 6.1: Schematic of a two-dimensional verification test case set-up. The vertical dashed line marks a typical profile for the evaluation of errors occurring errors along the dimensionless patch width $\tilde{L}$.

The induced error is usually evaluated relative to the reference properties using the L^2 relative error norm which reads

$$e_u = \frac{\sum_i^D (\boldsymbol{u}_i - \boldsymbol{u}_{ref,i})^2}{\sum_i^D \boldsymbol{u}_{ref,i}^2} \tag{6.2}$$

for the velocity error. As for the density, we normalise the deviation from the mean density fluctuation obtained from the test cases with the standard deviation of the density fluctuation found in the reference case along the referring profile which yields

$$e_\rho = \frac{|\rho - \bar{\rho}_{x^{eval}}|}{\sigma(\rho_{ref})} \quad , \tag{6.3}$$

where $\bar{\rho}_{x^{eval}}$ is the mean density fluctuation along the profile. The reason therefore is that even in the laminar flows used as a reference case we do not see a perfectly constant density fluctuation ρ along a profile. This variance can be interpreted as numerical noise, likely due to the computations running in single precision. Since the perturbation found in the reference cases is of a similar order as the perturbations found for the overset grids we shall apply the latter as a norm in order to put the induced errors into perspective.

As given above, we always apply Re $= 1000$, referring to $u_{ref} = 0.1\mathrm{m\,s^{-1}}$, simply to limit the number of test cases. Since the background flow is absolutely laminar we generally assume that the magnitude of background velocity is of minor importance for the final error of the method. For the sake of completeness, still, a brief investigation of this matter is provided by testing a two-dimensional patch with constant translational velocity at different Reynolds numbers but with the same velocity ration of $\Phi_p = 0.1$. The results are given in figure 5.7 as the relative velocity error found on the background patch along the dimensionless patch width $\tilde{L}$ at $x^{eval} = \boldsymbol{p}_x + L_f$ (at the easterly edge of the overset patch). Here, the dimensionless patch width is given by $\tilde{L} = y/L_c$. Note that for three dimensional cases y can also be replaced by z.

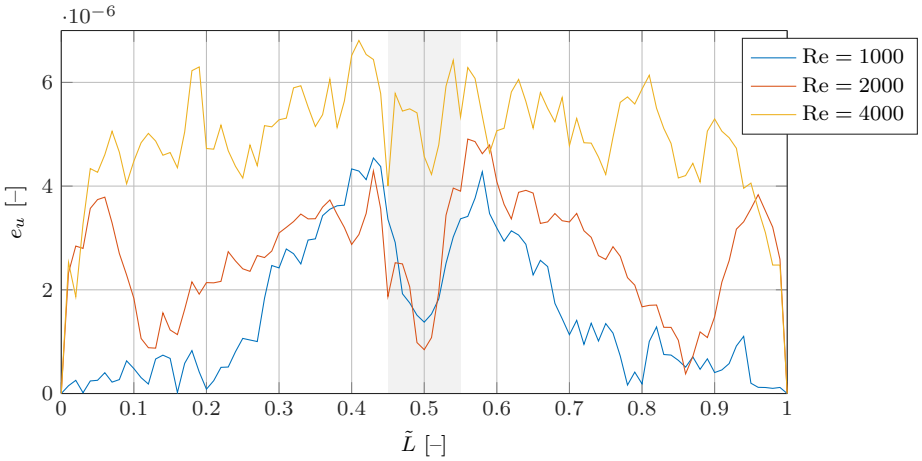


Figure 6.2: Relative velocity error induced by constantly moving patch with $\Phi_p = 0.1$ and Ma $= 0.05$ in flows of different velocity. The position of the overset patch is marked in light grey.

A certain dependency with respect to the flow velocity can obviously be observed. Yet, the errors lie closely within the same order of magnitude. The differences found for the investigated velocities are thus acceptable for validity of the following investigation.

6.1.2 Constant Linear Velocity

The most simple test scenario is an overset patch moving with constant linear velocity. Hence, there are no inertial forces acting in the moving reference frame, neither is the patch rotated. The entire transformation procedure falls back to the default case with both patches being aligned. Only the interpolation itself as well as the advection correction are applied when it comes to the overset procedures outlined in chapter 4.

As for this specific set-up the background patch refers to a channel of length $3\,L_c$. For all tested velocities the evaluation is performed along a profile at the same position, namely

$x^{eval} = 2\,L_c$ at time t such that the easterly edge of the overset patch coincides with x^{eval}. A Mach number of Ma = 0.05 is chosen.

To give a general impression of the perturbation caused by the overset grid the relative errors in the density fluctuation ρ and the velocity $\boldsymbol{u}$ are given in figure 5.8 for velocity ratios of 0.1, 0.25, 0.5 and 2.5. As we would encounter the same problems for $\Phi_p = 1$ as for tests with $u_{ref} = 0$ we do not consider this case in the evaluation. Furthermore, note that the case with $\Phi_p = 2.5$ is only shown as a bad example. Usually the reference velocity for a simulation should always be chosen to be larger than the maximum velocity on all grids, including the moving ones.

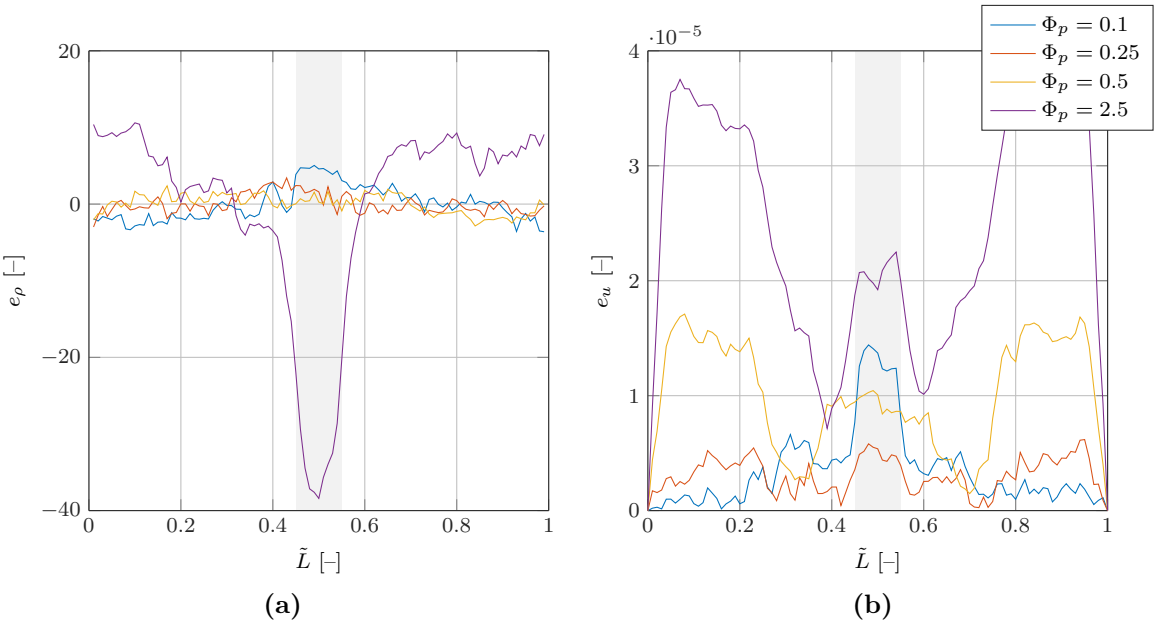


Figure 6.3: Relative errors in ρ (a) and $\boldsymbol{u}$ (b) induced by constantly moving overset patch at different velocity ratios.

Regarding the density fluctuation it can be seen that for $\Phi_p < 1$ the fluctuations lie within the same order of magnitude as in the reference case. A clear trend with respect to the patch velocity ratio can not be stated. As expected the errors for $\Phi_p = 2.5$ are significantly higher. Moreover, we should bear in mind, that ordinary flow induced density fluctuations in flows simulated with the incompressible LBM are generally of the order $\mathcal{O}(\mathrm{Ma}^2)$ He and Luo (1997). For this specific case with Ma = 0.05 and $\rho_{0,ref} = 1\mathrm{kg\,m^{-3}}$ a flow induced density fluctuation would thus refer to an error (using the above-mentioned norm) of the order $\mathcal{O}(10^6)$. Also, the errors found for the velocity do not show a clear trend with respect to the patch velocity. They are generally small for all investigated cases.

For further investigations we condense the occurring error and only consider the mean error $\bar{e}$ in the interval of $\tilde{L} = [0.45, 0.55]$ referring to the actual overset area. After all, errors found further away from the overset patch originate from this region. Using these metrics we perform the same evaluation for the overset method in three dimensions. The resulting mean relative errors for all test cases are shown in figure 5.9. It appears that the error generally shows a similar characteristic with respect to the velocity ratio but is

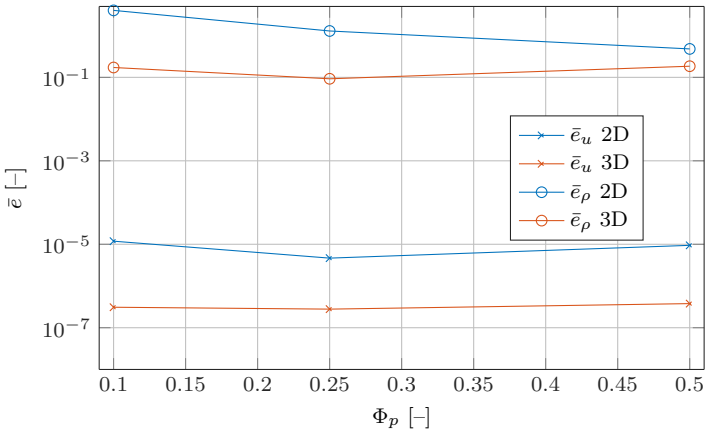


Figure 6.4: Comparison of mean relative errors in ρ and $\boldsymbol{u}$ induced by overset patch with constant linear velocity on two- and three-dimensional lattices.

found to be one to two orders of magnitude lower than in the two-dimensional cases. An explanation therefore can not be stated at this point, bearing in mind that the spatial resolution of the 3D case is identical to the 2D case when projecting the grid onto the x-y-plane.

6.1.3 Constant Linear Acceleration

Applying a constant linear acceleration alters the scenario of the above-mentioned test series by adding a constant linear inertial forcing. At this point, only accelerations parallel to the flow velocity are tested. The utilised set-up is identical to the one for the preceding test series and accelerations are chosen such that the maximum occurring velocity does not exceed u_{ref}.

With respect to the perturbation characteristics found for the different constant linear patch velocities no additional error can be clearly identified. This generally complies with the error term of the forcing applied to the LBE which only vanishes for $\boldsymbol{u}$ being parallel to the forcing (see section 2.5). For future test scenarios it is advisable to investigate cases of perpendicular accelerations, especially since the force-velocity coupling in the error term is still much simpler than for any rotating case.

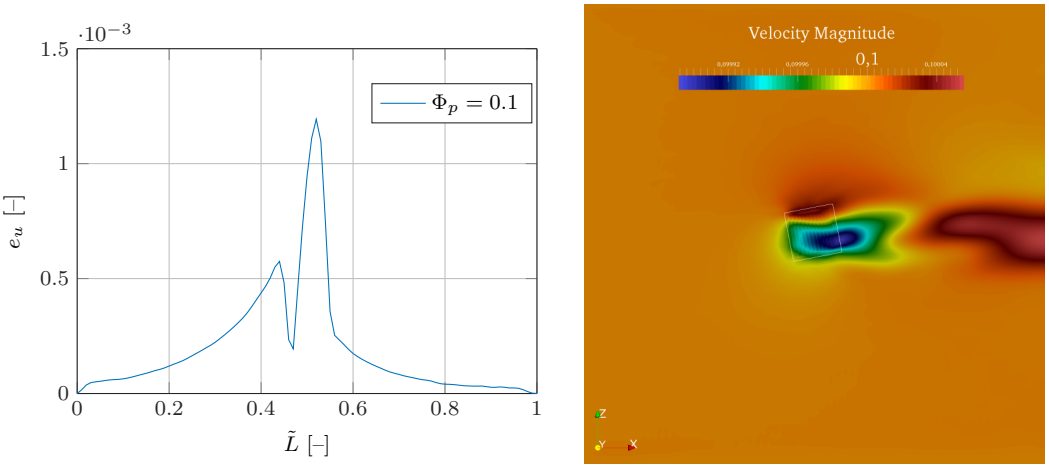
6.1.4 Constant Angular Velocity

Rotating a patch necessitates the application of all procedures developed within in this work. For the sake of brevity we only consider constant angular velocities in this test series. The direction of the additional Euler force arising from angular accelerations, though, coincides with the one of the centrifugal force. For this reason, specific errors due to this neglected parameter are not expected.

The overall test case set-up is again similar to the preceding cases but with the background patch only being L_c long. The centre of rotation of the overset patch coincides with its centre. The error evaluation is again performed along the profile at x^{eval}. The angular velocities are chosen in order to comply with the linear velocities investigated in

section 5.4.2 by means of the maximum circumferential speed with $u_{rot} = L_f/\sqrt{2}\,\dot{\varphi} = u_p$. The velocity ratio is defined with u_{rot} accordingly.

In contrast to the constant linear motion it is generally observed that the occurring error is depended on the orientation of the overset patch resulting in an oscillating error profile. Illustrative examples thereof are given in figure 5.10.



(a) **(b)**

Figure 6.5: Examples of oscillating error characteristics near rotating patches: (a) relative velocity error downstream of a rotating patch with distinct asymmetric characteristic. (b) Visualisation of the velocity in a two-dimensional domain with $u_{ref} = 0.1\,\mathrm{m\,s^{-1}}$.

Again, a condensed evaluation is performed equivalent to the one given for linearly moving patches, see figure 5.11. For the evaluation time steps are chosen where the easterly edge of the patch again coincides with the profile, referring to $\varphi = n\,\pi/4$. It appears that the error measured along the profile is usually largest at rotation angles like this. It is shown that the induced error in the velocity and density fluctuation is two to three magnitudes larger than for the constant linear motion. Also, the errors found on two- and three-dimensional lattices are very similar and the dependency on the velocity ratio is again generally small.

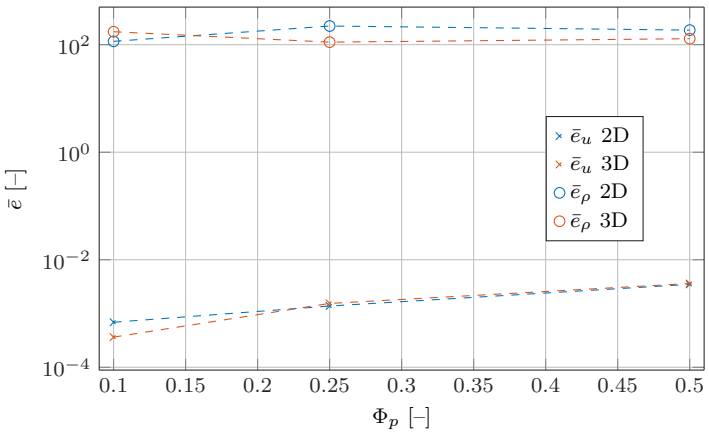
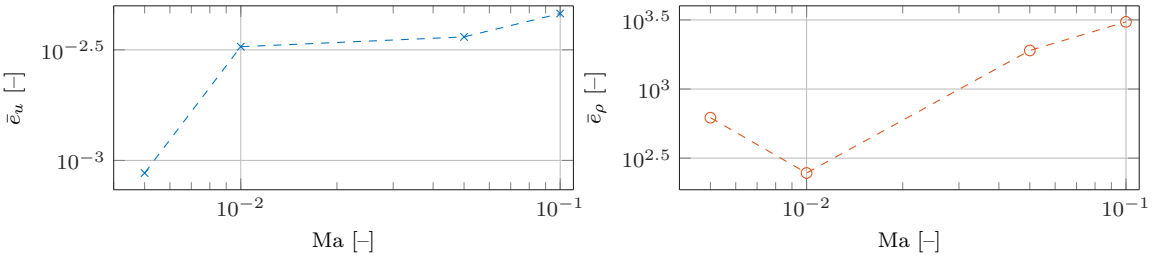


Figure 6.6: Comparison of mean relative errors in ρ and $\boldsymbol{u}$ induced by overset patch with constant angular velocity on two- and three-dimensional lattices.

Since the error caused by the forcing term scales linearly with the time step a further investigation is performed at $\Phi_p = 0.5$ with different Mach numbers. The mean relative errors are shown in figure 5.12. Even though first order convergence with respect to the Mach number can not be observed, the trend clearly indicates a decreasing error with decreasing Mach number.

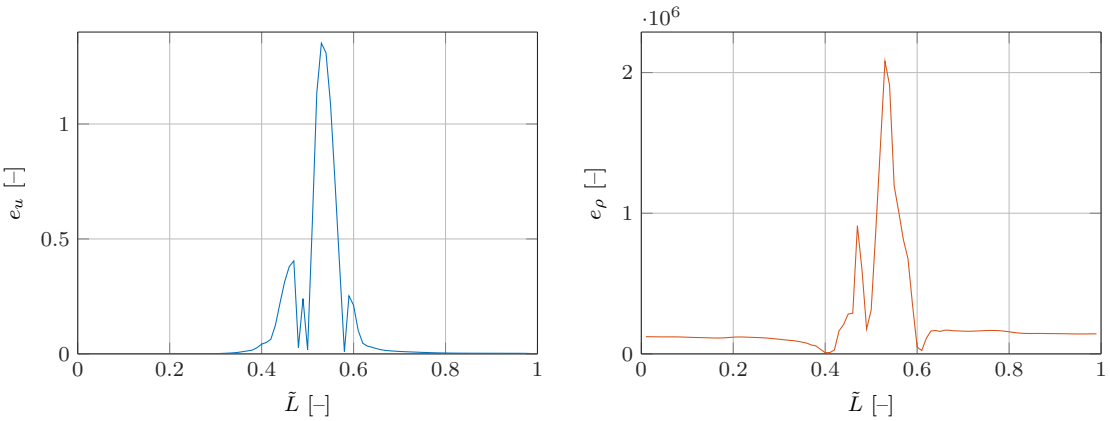


(a) **(b)**

Figure 6.7: Mean relative errors in $\boldsymbol{u}$ (a) and ρ (b) induced by an overset patch with constant angular velocity ($\Phi_p = 0.5$) at different Mach numbers on a three-dimensional lattice

.

All three-dimensional test cases presented here allow for a projection onto a plane and are thus compared to their two-dimensional counterparts. Lastly, we shall also give an example of a three-dimensional motion. Using the same set-up a rotation about more than one axis is applied. Figure 5.13 depicts the relative error profile of a three-dimensional patch rotating with constant speed about the x and y axis with $\dot{\varphi} = \dot{\theta}$. The velocity ratio for each component of the rotation is 0.1.



(a) **(b)**

Figure 6.8: Relative error profiles induced by overset patch with constant angular velocity about two axis.

It becomes obvious that the errors in $\boldsymbol{u}$ and ρ become prohibitively large. For the velocity the errors are of the order of the inflow velocity while the density errors caused by the overset patch reach the magnitude of flow induced density fluctuations of the order $\mathcal{O}(\mathrm{Ma}^2)$. Furthermore, it is found that the additional error caused by the second rotation increases linearly in the interval of $\dot{\theta} = [0, \dot{\varphi}]$.

6.2 Temporal and Spatial Convergence

In the preceding cases we have not discussed the influence of the spatial resolution with respect to the developed method in general. Since the considered background flows are spatially constant, we do not expect the spatial resolution to affect the quality of the results. For a further investigation of this topic we therefore investigate a Taylor-Couette flow where we observe a distinct velocity profile. Since we can determine an analytical solution for the laminar velocity of this case it perfectly suits the verification purpose as it allows for a straightforward evaluation of the steady state solution. Moreover, the nature of the Taylor-Couette flow allows us to simulate a moving object in ELBE without moving the object by means of the problematic node-updating. The rotation of the cylinders can simply be applied through a moving body boundary condition, in this case a LIBB scheme, see equation 2.32. The referring cylinder wall herewith refers to a classical slip boundary condition. Further to the analytical reference the results obtained from an overset set-up where the inner cylinder wall is an actual no-slip wall, can be compared to an ELBE reference case. This also allows us to compare the spatial convergence rates with and without overset grid. For the sake of completeness a temporal convergence study will also be conducted.

6.2.1 The Taylor-Couette Flow

The Taylor-Couette flow describes the fluid between two rotating cylinders. For $r_i/r_o \to 1$, with r_i being the radius of the inner and r_o being the radius of the outer cylinder the flow can be described by a simple Couette flow. Both numerically, and experimentally, this flow has been widely investigated within the last 60 years, often with respect to transition and stability of turbulent structures. Here, we shall only consider a purely laminar flow and make use of the available analytical description of the velocity profile. Compared to the preceding test case the Taylor-Couette flow has the major advantage of being in an enclosed domain with a constant solution along the rotation angle φ. Only comparing the velocity profile along the radius at an arbitrary φ consequently allows us to evaluate the quality of the entire solution.

Further to the above-mentioned radii we introduce the gap width d, with

$$d = r_o - r_i \tag{6.4}$$

as well as the inner and outer Reynolds number

$$\mathrm{Re}_i = \frac{\omega_i \, r_i \, d}{\nu} \quad \text{and} \quad \mathrm{Re}_o = \frac{\omega_o \, r_o \, d}{\nu} \quad , \tag{6.5}$$

where $\omega_{i,o}$ is the angular velocity of the referring cylinder. Moreover we define the radius ratio η as well as the shear Reynolds number Re_S following Nordsiek et al. (2015) with

$$\eta = \frac{r_i}{r_a} \tag{6.6}$$

$$\mathrm{Re}_S = \frac{2}{1+\eta} \left| \mathrm{Re}_i - \eta \, \mathrm{Re}_o \right| \quad . \tag{6.7}$$

Since grids in ELBE have to be rectangular, a minimum difference between the inner and outer radius has to be maintained in order not to overlap the overset grid with outer solid nodes. In this regime the curvature of the channel is not negligible and we obtain the actual Taylor-Couette azimuthal velocity profile which reads

$$u_\varphi(r) = A\,r + \frac{B}{r} \quad \text{, with} \quad A = \frac{\omega_o - \eta^2\,\omega_i}{1-\eta^2} \quad , B = \frac{r_i^2\,(\omega_i - \omega_o)}{1-\eta^2} \quad . \tag{6.8}$$

for low Re_S. The flow being laminar we obtain $u_r = 0$ for the radial velocity component. For this specific set-up we chose $\mathrm{Re}_i = 500$ and $\mathrm{Re}_a = -2000$ with the corresponding shear Reynolds number of $\mathrm{Re}_S \approx 666.67$. All underlying test case parameters are summarised in table 5.2. As we can see in figure 5.14 the chosen Reynolds numbers lie within the laminar

Table 6.2: Taylor-Couette test case parameters

Inner radius	r_i	[m]	1
Outer radius	r_o	[m]	2
Inner angular velocity	ω_i	$[\mathrm{rad\,s^{-1}}]$	0.05
Outer angular velocity	ω_i	$[\mathrm{rad\,s^{-1}}]$	-0.1
Shear Reynolds number	Re_S	[–]	666.67

Taylor-Couette regime. Also note that the simulation of turbulent Taylor-Couette flows in two-dimensional domains is generally not advisable as most typical turbulent structures are of three-dimensional character. An exemplary visualisation of the velocity vectors is

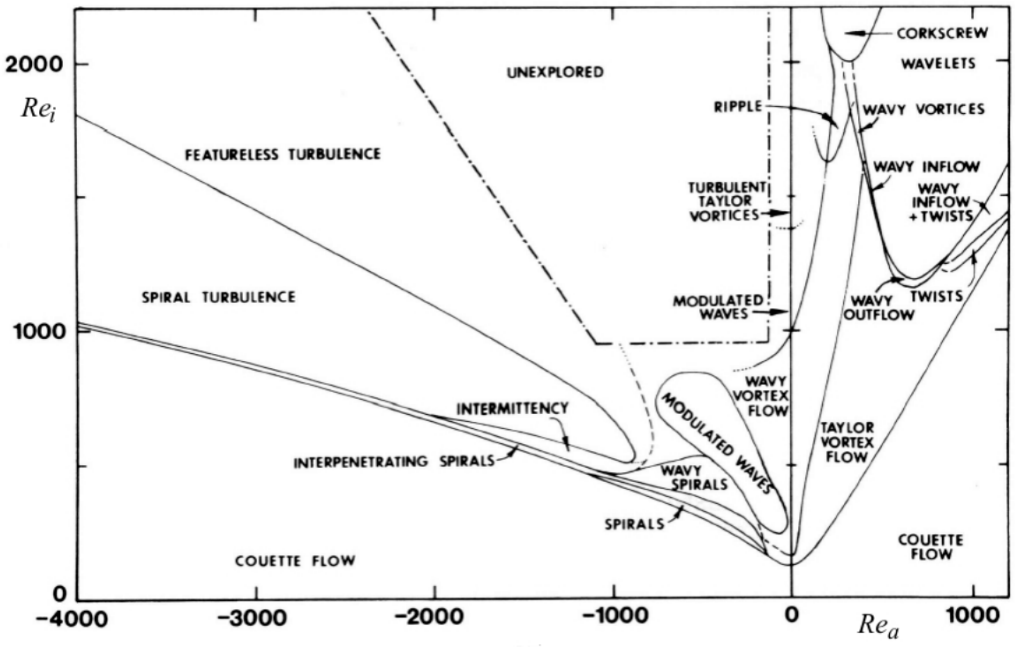


Figure 6.9: Experimentally determined stability diagram of Taylor-Couette flows. Source: Andereck et al. (1986)

given in figure 5.15. Here we can clearly see the resulting counter-clockwise rotation of the fluid near the inner cylinder and the clockwise rotation going further away from the centre.

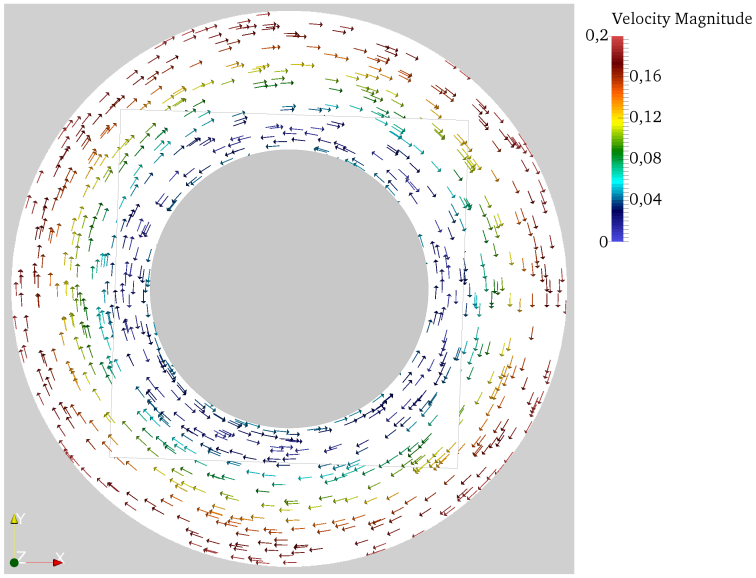


Figure 6.10: Exemplary velocity plot of a Taylor-Couette flow with overset grid. The inner grey box marks the overset patch including the inner cylinder.

Fur the following discussion we shall utilised the non-dimensional radius $\tilde{r}$ and the non-dimensional angular fluid velocity $\tilde{\omega}$ with

$$\tilde{r} = \frac{r - r_i}{d} \tag{6.9}$$

$$\tilde{\omega} = \frac{\omega(r) - \omega_o}{\omega_i - \omega_o} \quad \text{, with} \quad \omega(r) = u_\varphi(r)/r \tag{6.10}$$

Within the non-dimensional coordinates we consequently always obtain $\tilde{\omega} = 1$ at $\tilde{r} = 0$ referring to r_i and $\tilde{\omega} = 0$ at $\tilde{r} = 1$ referring to r_o.

For the determination of the convergence rates in the following section we apply the L^2-norm relative error norm of the velocity along $\tilde{r}$ using equation 5.10 for the reference velocity. The overall convergence will be determined using the spatial average of the magnitude of the relative error $\bar{e}_{u_\varphi}$.

6.2.2 Convergence Study in Ordinary and Moving Overset Grids

The LBM is theoretically second order convergent with respect to the velocity and pressure of the incompressible NS equations. It shows, however, that this only pertains for smooth

periodic flow. Even simple laminar flows including bounce-back boundaries usually only reach convergence orders of one and a half or less for the velocity Junk and Yang (2009a). Using acoustic scaling in ELBE second order convergence requires an equal scaling of space and time. Only scaling the spatial or temporal resolution can only yield first order convergence. For the overall comparison the overset case always implies a refinement of $s = 2$ around the inner cylinder. For the slip boundary case an equal node spacing is applied in the entire domain. The reference case with the slip boundary thus states the most simple set-up for this purpose without an interface interpolation or viscosity scaling.

Before discussing the actual convergence rates a qualitative comparison of the occurring error with and without overset grids is presented. Therefore we shall consider the velocity and relative error profile of the cases with the highest spatial resolutions at Ma=0.1. Results thereof are given in figure 5.16. Regarding the relative error we generally do not consider the domain between the wall at $\tilde{r} = \{0, 1\}$ and the first fluid node. Due to the bounce back schemes we can obviously only achieve a non-zero velocity at the first fluid node making an error analysis for the near-wall region obsolete. Figure 5.16a reveals

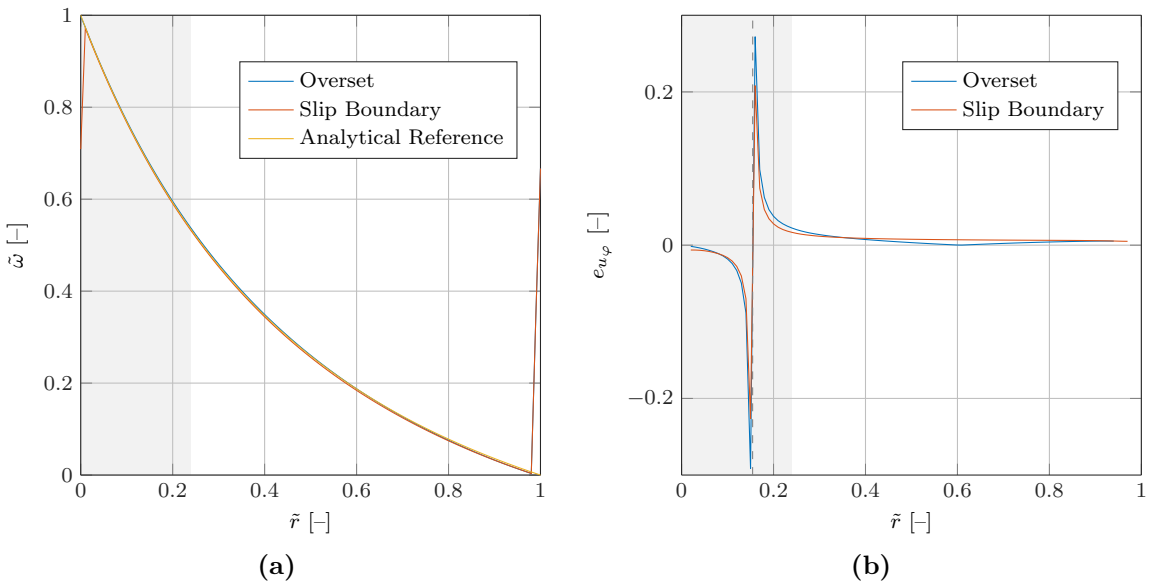


Figure 6.11: (a) Velocity and (b) relative error profiles of a Taylor-Couette flow with and without overset grid with a spatial resolution with 512 nodes per domain diameter (referring to the background patch for the overset case). The moving grid domain for the overset case is marked in grey. The dashed line in (b) marks the zero-crossing of the dimensionful azimuthal velocity.

that both approaches capture the analytical solution with very good agreement. Due to the zero-crossing of the velocity profile we find a singularity in the relative error profile at $\tilde{r} = 0.155$. Even though the overset case has a higher spatial resolution around the inner cylinder the error at the singularity is generally higher than in the reference case. Yet, the mean relative error agrees very well, also see figure 5.18. For the overset case it is important to remark that no distinct difference in the error profile can be observed between the overset and the background grid. The error is continuous across the grid interface.

Going into more detail, we also take into account the radial velocity component of the

overset cases and compare the solutions found for different spatial resolutions, see figure 5.17a. With increasing $\tilde{r}$ an increase of the erroneous radial velocity can be observed on the overset patch that also propagates further outwards through the background grid. Higher spatial resolutions tend to decrease the radial velocity. Even though the error with lower resolutions is also higher near the outer domain boundary the singularity clearly dominates the mean relative error, also see figure 5.18b.

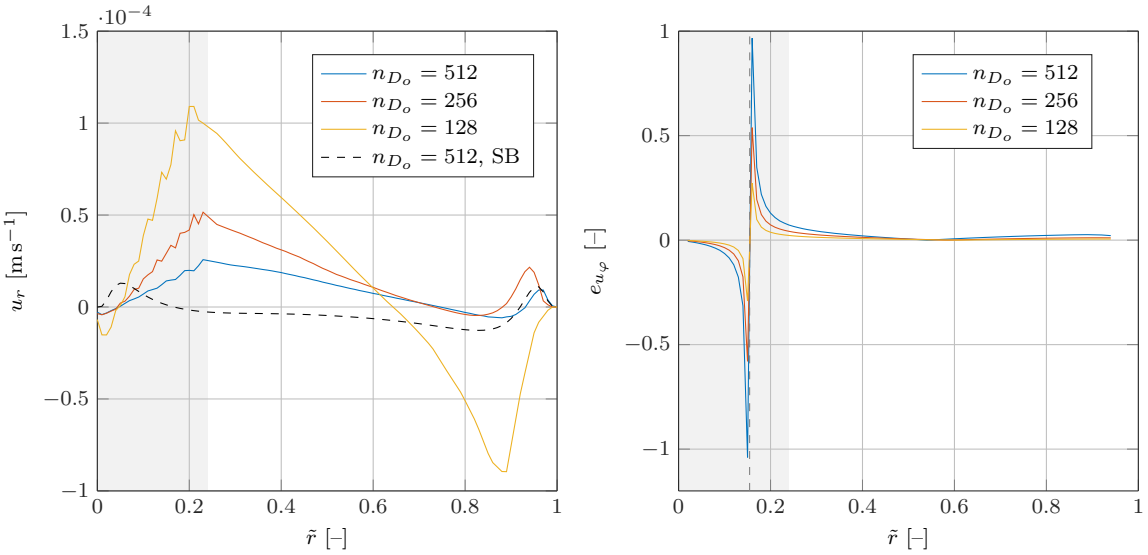


Figure 6.12: Comparison of the radial velocity (a) and relative error profiles (b) of a Taylor-Couette flow with overset grid with different spatial resolutions at Ma=0.1. For the sake of comparison the radial velocity profile of a slip boundary (SB) reference case is given additionally.

The radially increasing radial velocity suggests to be caused by the inertial forcing. For this test case the inertial forcing is dominated by the radial centrifugal force explaining the observed increase. Furthermore, the reference case, see figure 5.17a, does not show such a high radial velocity.

For the convergence study each set-up, the slip boundary and the overset case, are tested with three different spatial resolutions and with three different Mach numbers. The spatial resolution is given in nodes per domain diameter n_{D_o}. The results are depicted in figure 5.18.

For the slip wall case we can observe a spatial convergence of about first order complying with the theoretical expectation. Temporal convergence, however is not obtained. For all investigated resolutions we can observe a divergence with decreasing Mach number. For the overset case the spatial convergence is generally less than first order. The temporal convergence is also not achieved as all cases of Ma = 0.025 perform generally less accurate than with Ma = 0.05. Moreover, for Ma = 0.025 we find a kink with the error found for $n_{D_o} = 512$ being almost of the same magnitude as for the lower resolution.

A further investigation on the temporal divergence of the overset set-up reveals that the solution in the entire domain diverges for all Mach numbers smaller than 0.05. The latter is shown in figure 5.19. For Ma = 0.01 the solution is even worse than for Ma = 0.1 with the highest errors found around the zero-crossing.

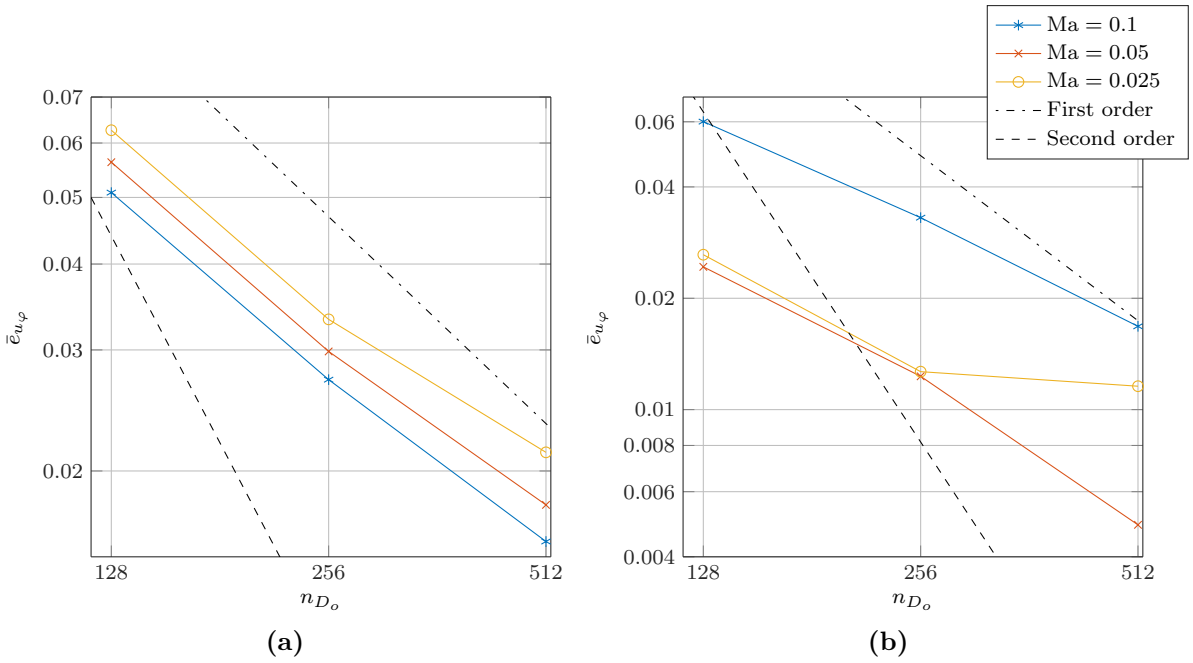


Figure 6.13: Spatial and temporal convergence study on a stationary grid with slip wall boundary conditions (a) and with overset grid (b) of a Taylor-Couette flow of $\mathrm{Re}_S = 666.67$. Exemplary first and second order rates are shown for guidance.

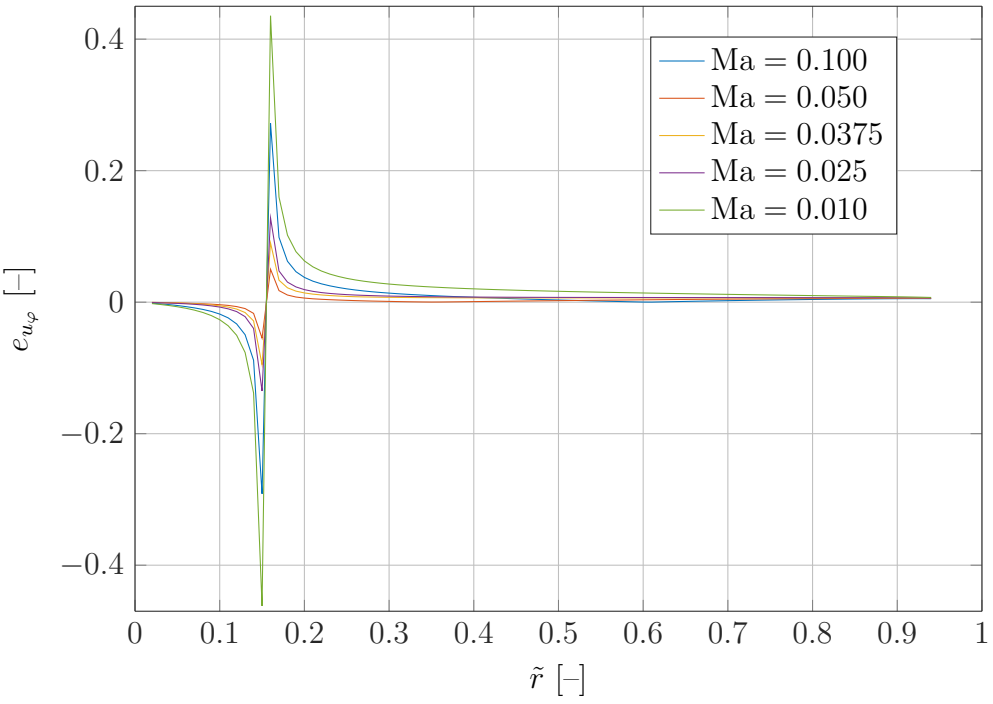


Figure 6.14: Relative error profiles of a Taylor-Couette flow with overset grid ($n_{D_o} = 512$) at different Mach numbers.

The resulting mean relative errors are given in figure 5.20 highlighting the divergence mentioned above. For comparison the behaviour of the corresponding slip wall case is

given additionally even though no solutions with further Mach numbers are computed.

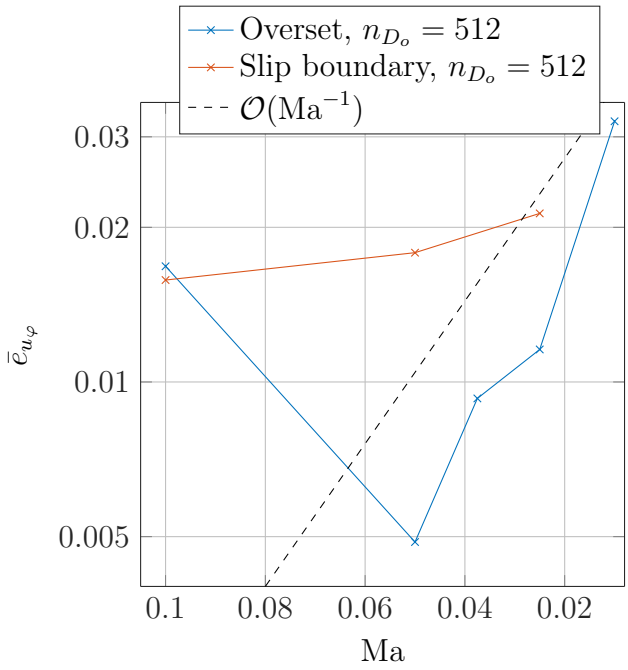


Figure 6.15: Mean relative error over Mach number of a Taylor-Couette flow with and without overset grid ($n_{D_o} = 512$). The dashed line shows an exemplary divergence of the order $\mathcal{O}(\mathrm{Ma}^{-1})$.

Even though the observed temporal divergence objects the incompressible low Mach number limit similar phenomena are reported in the literature. Dellar (2003) for instance presents several numerical experiments indicating a divergence at low Mach numbers of the order $\mathcal{O}(\mathrm{Ma}^{-1})$, similar to the one found here for the overset case, see figure 5.20. The author relates the divergence to the choice of relaxation times of the non-hydrodynamic moments in the MRT. It is suggested to include a Mach number dependence in the relaxation times of these modes in order to suppress an undesired coupling of the hydrodynamic modes directly related to the NS equations and the non-hydrodynamic additional modes. He also shows that the Mach number where divergence starts to occur is depended on the grid resolution. For the case shown here this might explain why all solutions of the slip wall case are divergent with respect to the time step size. After all, the overset cases are refined with a factor of two in the area around the zero-crossing which is shown to be dominant for the divergence of the entire solution. The latter can be corroborated by further investigations simply expanding the scope of grid resolutions and Mach numbers. Alternatively, the overset case can be set up without refinement.

6.3 Discussion and Concluding Remarks

The cases presented within this verification test series provide an overall error estimation that is acceptable within the limits of expectations of the applied numerical methods. The developed transformation method as well as the advection correction can be considered as verified. Concluding from the errors determined within the different scenarios the main problem seems to be associated with the utilised forcing term. Particularly, this becomes obvious comparing test cases of constant linear velocity and constant angular velocity and then again with cases of superposed rotations. Here, it is striking how drastic the increase

of the error becomes despite the fact that the relative velocities of the patch are kept within the same order of magnitude for all of the afore-mentioned cases. Taking a closer look at the error term of the forcing, however, reveals that such a non-linear increase can be expected. We can rewrite the additional erroneous forcing term given in equation 2.36 in a component-wise formulation, and obtain

$$\Delta F_\alpha = \left(\tau - \frac{1}{1}\right) \Delta t \sum_\beta \partial_\beta (\boldsymbol{u}_\alpha \boldsymbol{F}_\beta + \boldsymbol{u}_\beta \boldsymbol{F}_\alpha) \quad . \tag{6.11}$$

Now considering the change from one rotation to two rotations of the same magnitude, yields additional velocity and forcing components in one spatial dimension, meaning three instead of two. As for the forcing error it follows that we add three summands in all vector components instead of two in only two vector components while additionally increasing the spatial derivatives of both, velocity and forcing due to the second rotation. Only based on this qualitative analysis of the error term we can certainly conclude that the applied forcing term is simply unsuitable for the application of spatially complex inertial forces. Not only to allow for multiple superposed rotations but also to decrease the error found for simple rotations the implementation of other forcing terms should definitely be considered.

The spatial convergence of the overset case is found to be less than the expected first order which again is found for the Taylor-Couette reference case. The scope of the conducted convergence study leaves us with the question to which component of the overset method the lower convergence is related. Again, a relation to the forcing term suggests itself since the linear interpolation does not reveal a distinct local influence on the solution. Further to the quantitative convergence study we do refer to a rather qualitative observation made in the verification study in conjunction with the spatial resolution, namely the occurrence of checkerboarding effects on the overset patch. These can be observed for overset patches independent of their kinematics, even for stationary ones, see e.g. figure 5.21. Moreover,

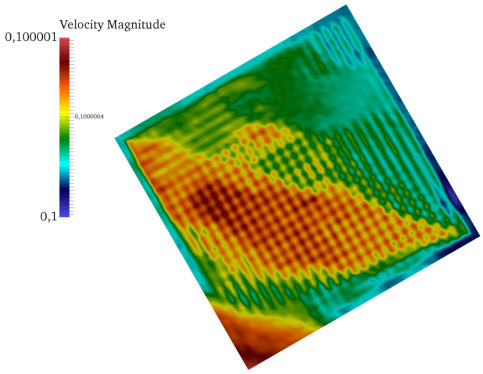


Figure 6.16: Example of checkerboarding effects on a stationary rotated overset patch.

it shows that increasing the spatial resolution reduces this effect. Based on the present findings on moving and stationary patches it can be suspected that the checkerboarding is reasoned by the linear interpolation applied at the patch interface. Here it is especially striking that the effect, if present, can not be observed directly at the receptor nodes of the overset patch but on the rest of the local domain. At this point one might suggest that this is due to the insufficient order of interpolation of higher order moments. Future

investigations will have to clarify whether this is actually the case or if any other part of the applied method is causing this numerical degeneracy. A first step could be the implementation of the complex interpolation by Geier et al. (2009a) or any simple higher order polynomial formulation, even if only for the sake of investigation.

For the investigated cases in the convergence study, both with and without overset grids, a divergence at low Mach numbers is found. Based on the fact that both cases show this characteristic as well as the above-mentioned reference it is not suspected that this phenomenon is related to the overset method. It rather seems to expose a general issue, possibly related to the MRT collision operator.

To conclude, the verification study revealed certain accuracy issues of the developed overset method. Nevertheless, none of the observed remains unrelated to expected issues and therefore consolidates the focus for future improvements.

Chapter 7

Validation and Application

Consulting two- and three-dimensional test cases we shall validate the developed method for complex flows and, in particular, with moving bodies being represented by the overset grid. The focus hereby not only lies on the comparison to peer-reviewed benchmark cases but also to reference cases computed in ELBE. Test results with and without overset grids compared to benchmark cases will be used to evaluate to what extent moving overset grids improve the simulation of moving bodies in ELBE.

7.1 Two-dimensional Oscillating Cylinder Test Cases

The presented two-dimensional test case is chosen in order to highlight different aspects of overset grids in comparison to stationary grid approaches. The oscillating cylinder test case describes a complex, turbulent flow where node-updating can not be avoided without overset grids. This test case has been extensively studied by Mierke (2015) evaluating the accuracy of different bounce-back schemes in combination with node-updating. It consequently states a perfect case to determine the benefits arising from overset grids.

The study of vortex-induced vibrations of a circular cylinder by Mittal and Kumar (1999) states a widely used benchmark case for flows around moving bodies. The authors apply a space-time finite element method to study a cylinder at $\mathrm{Re} = 325$ that is mounted on a spring-damper support. The support allows for in-line and cross-flow vibrations of the cylinder about its equilibrium position. The interaction of the oscillatory vortex shedding with the motion of the cylinder thus states a perfect example for a bidirectional fluid structure interaction.

As mentioned above, Mierke (2015) performed a wide range of parameter studies of this case investigating the effects of different bounce-back schemes as well geometry discretisation methods with respect to accuracy. It is shown that the differences between SBB and higher order bounce back schemes are significant. Discrepancies of the cylinder's trajectories when compared to the reference data, even with LIBB or QIBB and high spatial resolutions of the domain, are, however, noticeable. Here, a set-up with LIBB and an analytical geometry representation (as opposed to a polygon representation) shall serve as an ELBE reference case. The chosen set-up with a spatial resolution of 32 nodes per diameter D refers to the best matches with the benchmark case as presented by Mierke (2015).

The oscillatory vortex shedding behind the cylinder can be described by the Strouhal

number with

$$\mathrm{St} = \frac{f\,D}{u} \quad , \tag{7.1}$$

where f is the frequency of the vortex shedding. For a cylinder at $\mathrm{Re} = 325$ one finds $\mathrm{St} = 0.21$. In addition to that we define the following non-dimensional parameters for this test case: The non-dimensional cylinder mass

$$\bar{m} = \frac{4\,m}{D^2\,\rho} \quad , \tag{7.2}$$

the non-dimensional natural frequency of the support

$$F_s = D\,\frac{\sqrt{k/m}}{2\pi\,u} \quad , \tag{7.3}$$

where k is the spring stiffness and m the mass of the cylinder, and, lastly the structural damping coefficient

$$\zeta = \frac{c}{2\,m\sqrt{k/m}} \tag{7.4}$$

with the dimensional damping constant c. Hence, for $F_s = \mathrm{St} = 0.21$ the system becomes resonant describing the most severe motion of the cylinder. $F_s = 0.20$ as well as $F_s = 0.42$ will be investigated in this work. For the other parameters we choose $\bar{m} = 4.7273$ and $\zeta = 0.00033$. For the force calculation the MEM method is applied. The spatial resolution is set to 32 nodes per diameter for the LIBB case and 64 nodes per diameter for the overset case. Note that despite the different resolutions the solution in both cases is fully converged with respect to the grid resolution. Grid independence is thus ensured implying that both solutions are comparable despite the different resolution. The overall domain measures $32\,D$ in length and $16\,D$ in width. The overset patch is quadratic and with $L = 8\,D$.

A velocity plot of the domain with overset patch is shown in figure 6.1. The plot shows the instantaneous velocity field after seven seconds. At this point the cylinder is moving on a constant trajectory. We can observe the typical oscillating vortices in the wake of the cylinder decaying further downstream. Regarding the overset patch no distinct perturbation of the flow field is noticed.

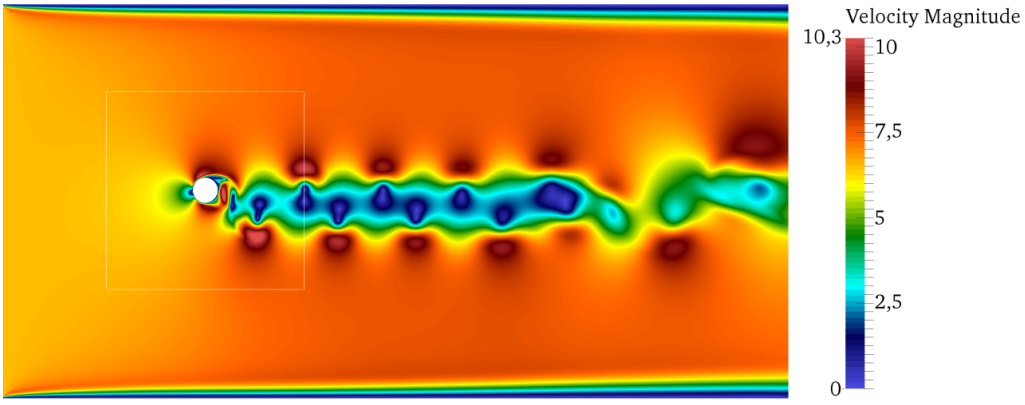


Figure 7.1: Velocity plot of vortex-induced oscillating cylinder with overset patch.

The resulting trajectories of the cylinder displacement normalised with the cylinder radius r are given in figure 6.2. For $F_s = 0.20$ which is close to the resonant state the

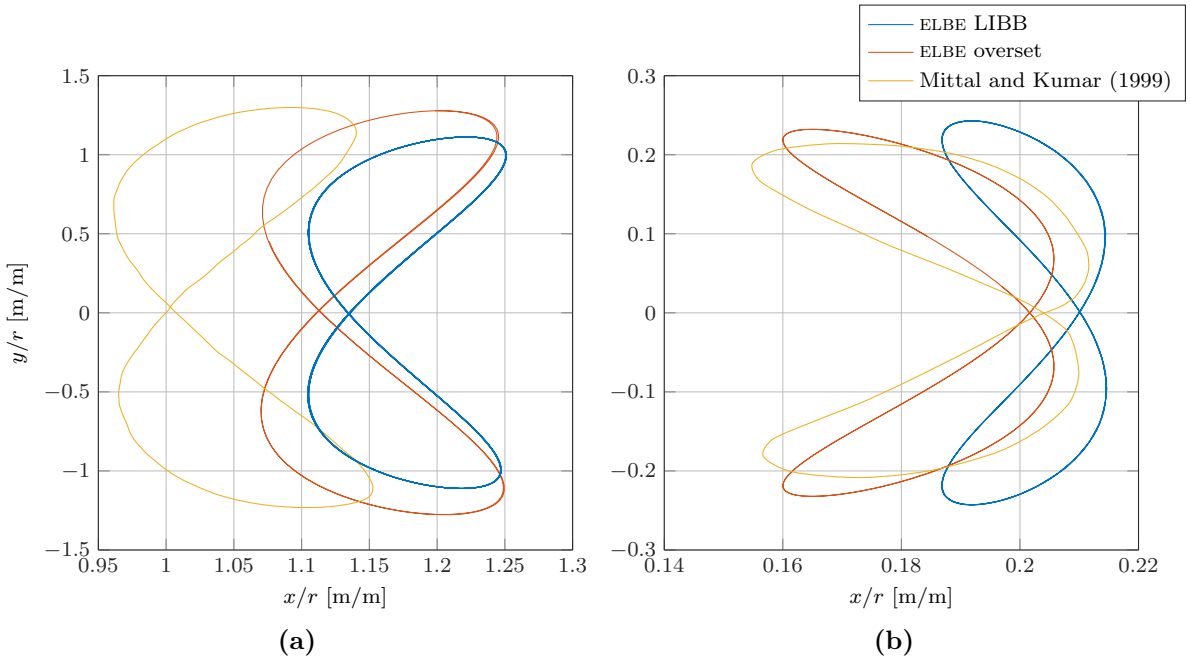


Figure 7.2: Comparison of the trajectories of vortex-induced oscillations of a cylinder with LIBB and overset grid for $F_s = 0.20$ (a) and $F_s = 0.42$ (b).

amplitude of the cross-flow oscillation with overset grids matches the reference case significantly better. Generally the characteristic of the trajectory matches better than for the LIBB case. Still, a significant offset of the mean displacement of 10.94% can be observed. For the LIBB case the displacement error amounts to 13.14%. Similar observations are to be made for $F_s = 0.42$. Again, the overall characteristic of the trajectory matches the reference better. Here, the mean displacement of the overset grid case deviates from the reference by only 1.18%. The LIBB case shows a mean displacement error of 2.99%.

To conclude we can generally say that major improvements with respect to the reference case are achieved by applying overset grids to ELBE. We can assume that the remaining differences between the overset cases and benchmark data is again related to the accuracy issues outlined in chapter 5.4. Even though the overset patch only performs a linear motion, no rotation, the applied forcing can be expected to cause serious errors due to the oscillating flow and especially the cross-flow motion of the patch. Furthermore, the solution of this case is very sensitive to the accuracy of the force evaluation method as the motion of the cylinder is directly coupled to the computed forces.

7.2 Three-dimensional Test Cases

For the three-dimensional test cases technical applications are chosen leaving the field of simplified academic benchmark cases as for the cases in two dimensions. First is the Potsdam Propeller Test Case (PPTC), an experimental open water test series of a model scale five-bladed controllable pitch propeller. Second is the simulation of a Voith-Schneider

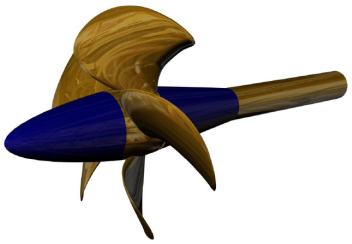
propeller, a cyclorotor consisting of five blades mainly used for tugboats. For the former, again, a quantitative analysis is provided comparing the results to an ELBE reference case as well as the aforementioned physical experiments. Due to the limitations of this work, the latter shall only provide a qualitative comparison of improvements gained from overset grids.

7.2.1 The Potsdam Propeller Test Case

The PPTC (Barkmann, 2011) comprises a series of experiments of the VP1304 propeller. Here, we consult the open water test series providing detailed data sets of the torque and thrust of the propeller. The entire test series is provided by the Schiffbauversuchsanstalt (SVA) for the purpose of validation of numerical simulations in the field of ship propulsion. In the open water tests the propeller is placed in a towing tank and exposed to a constant laminar flow. The thrust and torque are measured with a dynamometer placed behind the propeller model. The final results include corrections for the resistance of the hub and shaft as well as the additional friction of the bearings. A summary of all test case specifications is given in table 6.1.

Table 7.1: PPTC propeller specifications and test parameters as well as CAD drawing of the VP1304 propeller, Source: Barkmann (2011).

Diameter	D	[m]	0.25
Pitch ratio $r/R = 0.7$	$P_{0.7}$	[–]	1.635
Chord length $r/R = 0.7$	$c_{0.7}$	[m]	0.10417
Skew	θ_{EXT}	[°]	18.837
Number of blades	N	[–]	5
Number of revolutions	n	[s^{-1}]	10
Flow velocity	u_a	[$m\,s^{-1}$]	2



A previous work by Angerbauer (2015) also validates ELBE with the experimental data of the PPTC. Here, however, a different set-up is chosen. Besides the standard approach for moving geometries using a bounce-back scheme with node-updating a globally rotating fluid domain is applied. In the domain the propeller is not moving while the fluid is given a rotation using velocity boundary conditions as well as a corresponding inertial forcing. The situation in the overset domain around the propeller is thus very similar to the case by Angerbauer (2015). The main difference states the interpolation and transformation at the subdomain boundary. We therefore assume that the solution found with the overset setup should perform worse with respect to accuracy than the case with the globally rotating domain. Additionally, two constraints need to mentioned that might affect the quality of the overset results. First, at the point of writing the implemented connectors do not explicitly cover the case of solid nodes residing within the grid interface. As a result thereof, the shaft of the propeller has to end within the overset patch, see figure 6.3. The set-up by Angerbauer (2015) maps the geometry up to the outlet of the domain which definitely complies better with the upstream configuration of the actual SVA propeller.

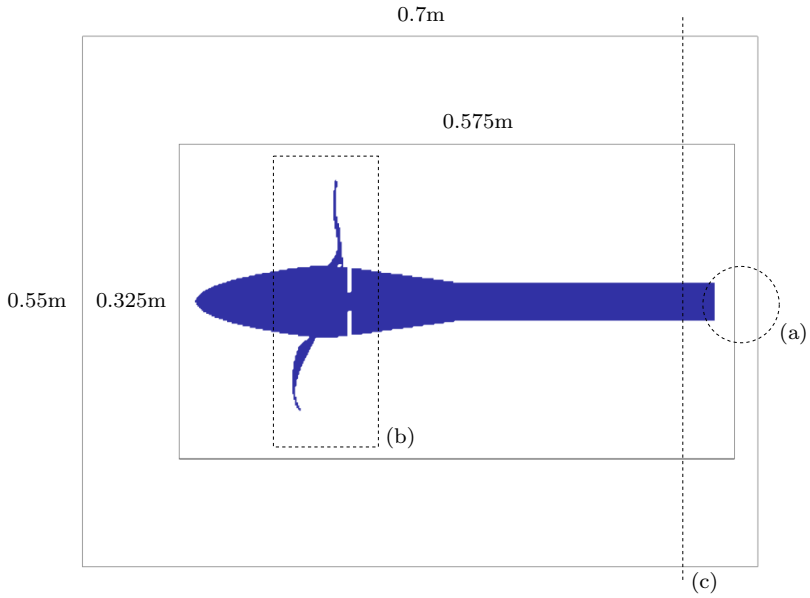


Figure 7.3: Domain set-up of the PPTC. The inner rectangle marks the boundary of the overset patch. The propeller geometry is solely mapped into the overset domain leaving a gap of fluid between the shaft end and the outlet, see (a). The dashed rectangle (b) marks the additional refinement area included in the reference set-up. (c) indicates the domain boundary of the reference case where the shaft is cut by the outlet boundary.

Secondly, the current implementation does not allow for a further refinement within a moving overset patch. After all, such would refer to an ordinary, stationary refinement if the refined patch within the moving overset patch complies with the same moving reference frame. As a consequence a second refinement zone around the propeller blades is not possible. The maximum number of nodes per diameter is thus limited to a about 205. This limit is determined by the internal memory capacity of the available GPUs with the corresponding maximum number of nodes per computation. Note that Angerbauer (2015) used up to 390 nodes per diameter as a second refinement could be applied.

The evaluation of the quality of the results is conducted using the non-dimensional torque and thrust coefficients of the propeller

$$k_Q = \frac{Q}{\rho_0 \, n^2 \, D^5} \quad \text{and} \tag{7.5}$$

$$k_T = \frac{T}{\rho_0 \, n^2 \, D^4} \quad , \tag{7.6}$$

where T is the total thrust, Q the total torque, n the number of revolutions and D the propeller diameter. For the open water diagram of the propeller the torque and thrust coefficients are plotted against the advance coefficient with

$$J = \frac{u_a}{n \, D} \quad . \tag{7.7}$$

Moreover, the Reynolds number is defined according to Barkmann (2011) as

$$\mathrm{Re}_P = \frac{c_{0.7}}{\nu} \sqrt{u_a^2 + (0.7 \, D \, \pi \, n)^2} \quad . \tag{7.8}$$

Initially, a brief spatial convergence study is performed. Without further testing, a Mach number of 0.05 is chosen based on the findings by Angerbauer (2015). The spatial convergence is evaluated using the relative error e of the thrust and torque coefficient at an advance coefficient of $J = 0.8$. The thrust and torque coefficients are determined as the mean value over the last of five simulated revolutions. With a number of revolutions of $10\,\mathrm{s}^{-1}$ this refers to an inflow velocity of $2\,\mathrm{m\,s}^{-1}$. The error is plotted against the number of nodes per propeller diameter N_D, see figure 6.4. The chosen configuration refers to a Reynolds number of $6.09 \cdot 10^5$. It becomes obvious that the highest resolution shows very

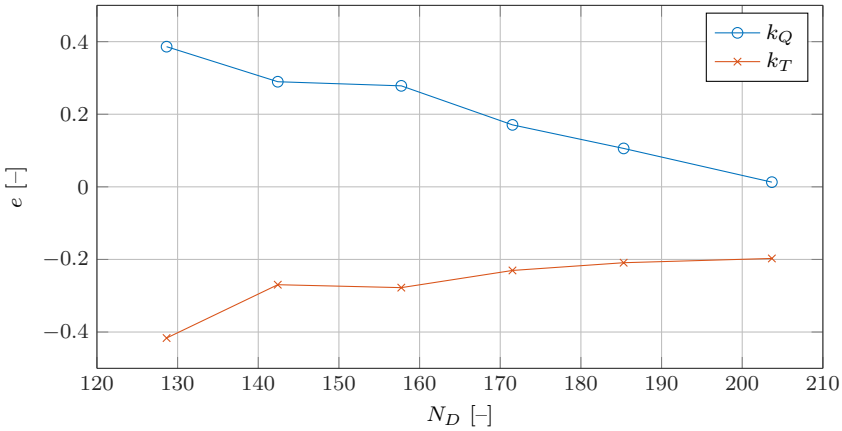


Figure 7.4: Spatial convergence study of the PPTC at $J = 0.8$ and $\mathrm{Ma} = 0.05$.

good agreement with respect to the torque coefficient while the thrust coefficient is still underestimated by about 20%. The large difference between the two error quantities is assumed to be caused by the back flow at the end of the propeller shaft occurring due to the above-mentioned gap, also see figure 6.5. As the resolution also refers to the maximum number of nodes possible within the current set-up all further computations are run on this grid.

The velocity field, see figure 6.5, reveals that even highly turbulent three-dimensional flows are continuously resolved across the grid interface. Moreover, the above-mentioned recirculation becomes clearly obvious at the shaft end. Near the subdomain boundary of the overset patch horizontal distortions of the velocity field are to be seen. Similar phenomena are reported by Angerbauer (2015). The author relates their occurrence to the high relative velocities in these regions that exceed the velocity limit determined by the macroscopic reference velocity. The flow is thus temporally under resolved. In order to maintain a reasonably low temporal resolution the author applies an increased artificial viscosity at radii $r > 0.6\,R$. Here, however, the reference velocity is chosen as the maximum relative velocity on the overset patch implying an appropriate temporal resolution. It is therefore concluded that the occurring disturbances are rather related to the forcing formulation.

The final open water diagram of the propeller is shown in figure 6.6. The torque coefficients generally matches the reference data very well throughout the entire range of tested advance coefficients. The only outlier is found for $J = 0.6$. Interestingly, also the results by Angerbauer (2015) show a similar, yet less severe, trend in that area. The thrust coefficient is almost constantly under estimated as already discussed in the context

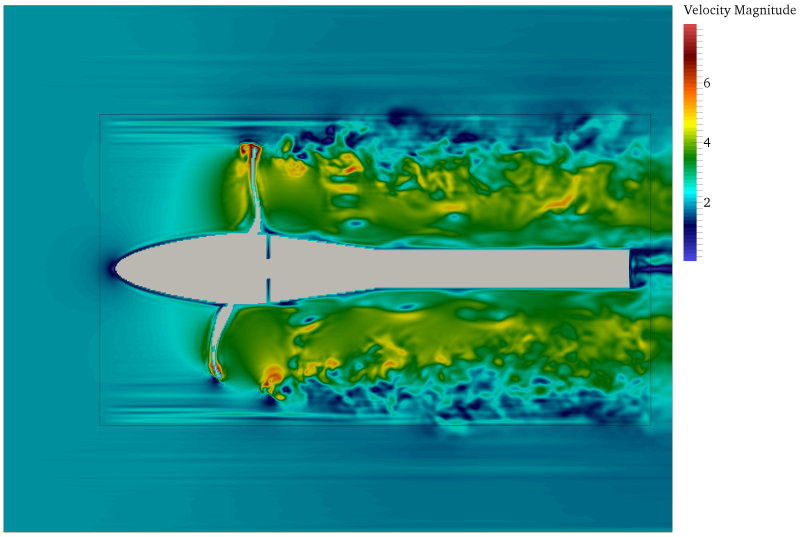


Figure 7.5: Cross sectional view of the velocity field of the PPTC at $J = 0.8$.

of the convergence study. For $J < 0.4$ a significant kink can be observed. It is most probably related to the small domain in conjunction with the inflow velocity being small compared to the velocity induced by the propeller.

We can conclude that despite the restrictions on the utilised test case set-up the results obtained stand in good agreement with the reference data. As we can assume that the gap at the end the shaft has less effects on the torque than on the thrust an evaluation of the result quality should rather be based on the former. The quality of the results is particularly striking bearing in mind the comparably low utilised spatial resolution. After all, the artificial viscosity and lower temporal resolution in the ELBE reference case are the main differences regarding the set-up. We can therefore assume that an appropriate temporal resolution is crucial for the final quality of the results. More generally it can be concluded that the developed overset method can be considered as validated for highly turbulent three-dimensional flows.

7.2.2 Simulation of a Voith Schneider Propeller

The Voith Schneider Propeller (VSP) is a typical cyclorotor that is commonly used for ships that require quick and versatile manoeuvrability. It consists of several vertical blades (usually four to five) that reside under the hull of a ship. The blades perform a rotation about the centre of the rotor whilst rotating about their own axis. The VSP allows for an arbitrary adjustment of the total rotor thrust and direction by adjusting the pitch characteristic of each individual blade. A ship with two VSPs can perform any possible motion, including sidewards. An exemplary trajectory of one blade as well as a photograph are given in figure 6.7.

The VSP states a perfect test case for the application of moving overset grids. At the point of writing no comparison of the rotor thrust or any flow parameter is available for the VSP. Also, the superposition of two rotations is simplified for the computation of the inertial forcing. Here one instantaneous linear acceleration is applied for the rotation

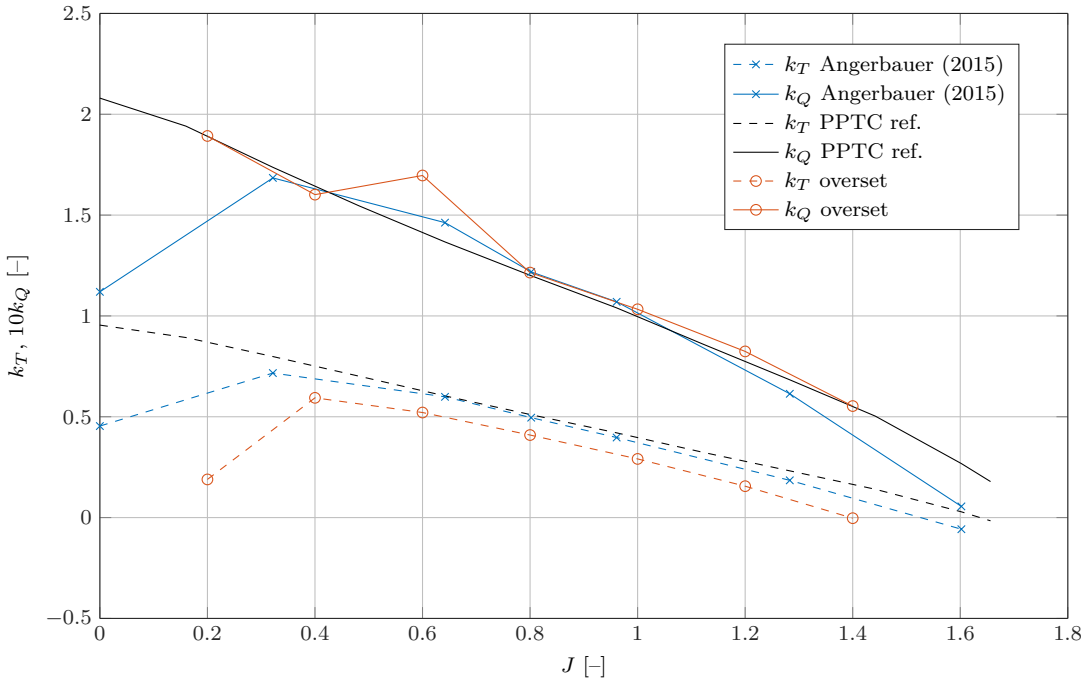


Figure 7.6: Comparison of open water diagrams of the PPTC propeller. The data by Angerbauer (2015) refers to the best fit obtained in the study using a resultion 392 nodes per diameter at Ma = 0.05. The PPTC reference data refers to the experimental results by Barkmann (2011).

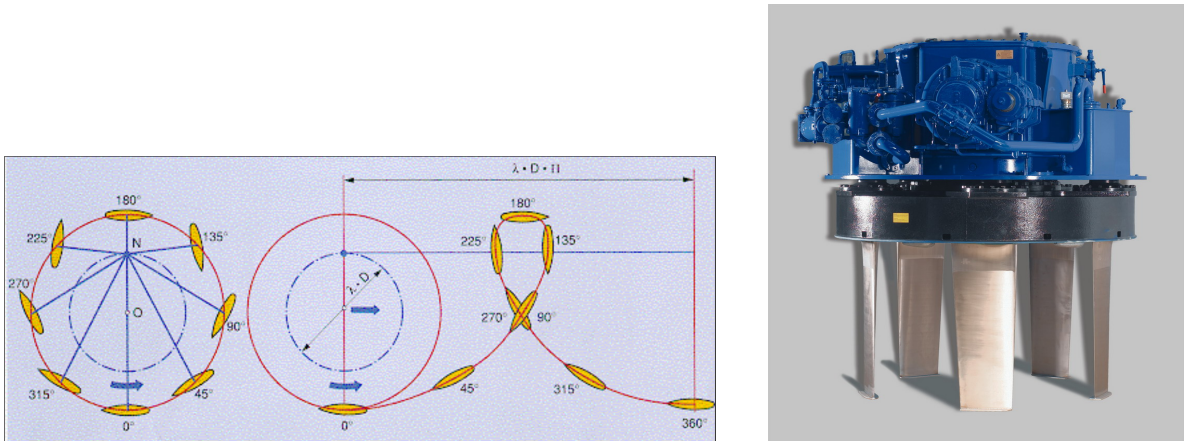
about and the rotor centre as well as the actual rotation of the blades about their own axis. Yet, a clear improvement of the simulation results compared to a standard ELBE set-up can be observed. The case shall thus rather be seen as an outlook for future application also giving the proof of concept for multiple moving patches on one background grid. The general test case parameters used here are given in table 6.2. For the sake of brevity it

Table 7.2: Voith Schneider Propeller test case parameters

Number of blades	N	[-]	5
Blade type	NACA0018		
Chord length	c	[m]	1.1
Rotor radius	r	[m]	1.6
Number of revolutions	n	$[\mathrm{s}^{-1}]$	1.18
Flow velocity	u_a	$[\mathrm{m\,s}^{-1}]$	6.23

is referred to Fork and Jürgens (2002) for a detailed description of the kinematics of the individual blades.

In the presented set-up a five bladed VSP is placed in a laminar inflow applied through velocity boundary conditions at all patch boundaries except the outlet. Here a pressure boundary condition is applied. In the overset case the blades are placed in moving patches with a cross section of $1.36\,c \times 0.5\,c$. The ELBE reference case applies the standard set-up using node-updating and a moving body LIBB boundary condition for the blades. An



(a) **(b)**

Figure 7.7: Explanatory illustrations of a Voith Schneider Propeller: (a) exemplary trajectory of one blade of a VSP during a global motion from left to right. Source: Wiki2 (2016). (b) Photograph of a VSP. Source: Voith AG.

exemplary visualisation of the overset set-up is given in figure 6.8.

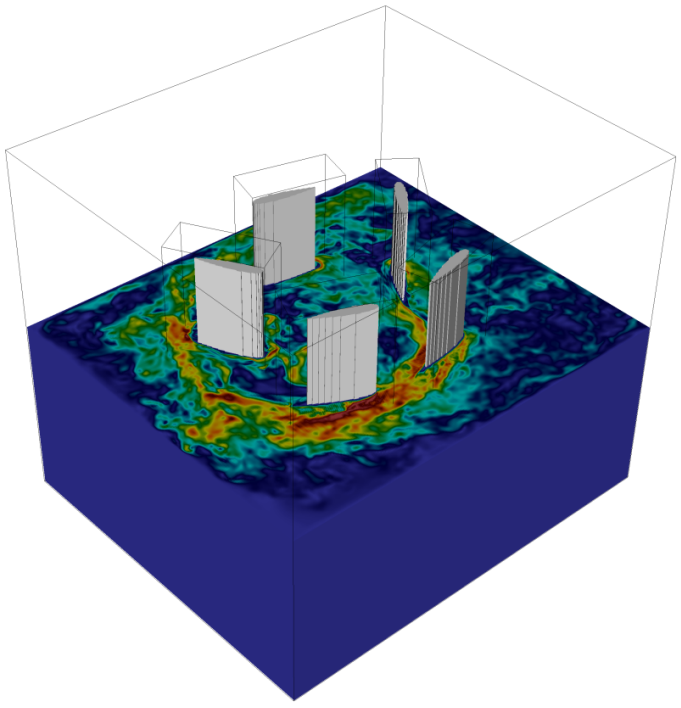


Figure 7.8: Exemplary visualisation of the VSP overset set-up showing five VSP blades each mapped into a moving overset patch. The black boxes mark the AABB of the referring patches.

First, we shall discuss the ELBE reference case, shown in figure 6.9. Throughout the entire domain we can observe the typical erroneous fluctuations. In some areas these only show as a light checkerboarding effect. However, in areas with high velocity gradients where fluid nodes are being reinitialised they occur as severe distortions of the velocity

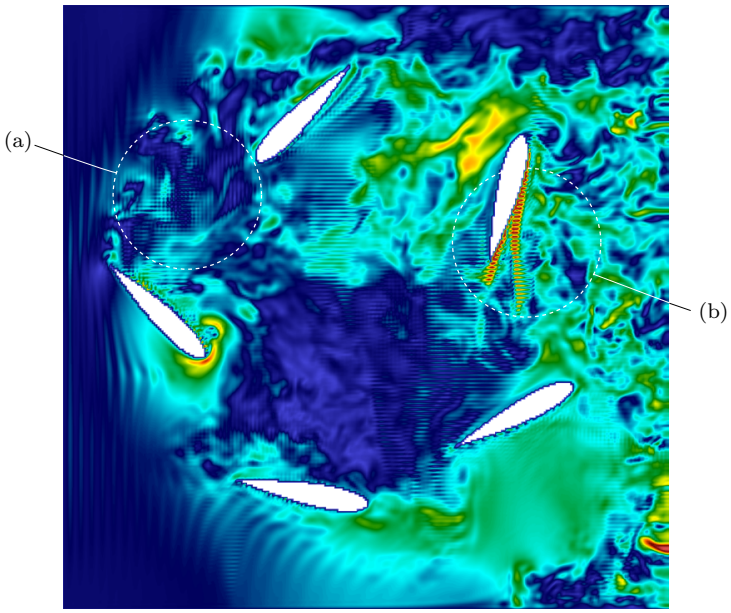


Figure 7.9: VSP velocity field with standard moving body approach. (a) marks light checkerboarding effects in the wake of a blade. (b) indicates the direct occurrence of erroneous node reinitialisations in the vicinity of a blade.

field. The propagation of these fluctuations obviously affects the entire domain. Even towards the inlet we can observe large oscillations.

An example of the corresponding overset case is given in figure 6.10. It can generally be stated that again a continuous velocity field is to been seen across the patch interfaces. Severe oscillations can not be observed. Yet, we can see light checkerboarding effects in the

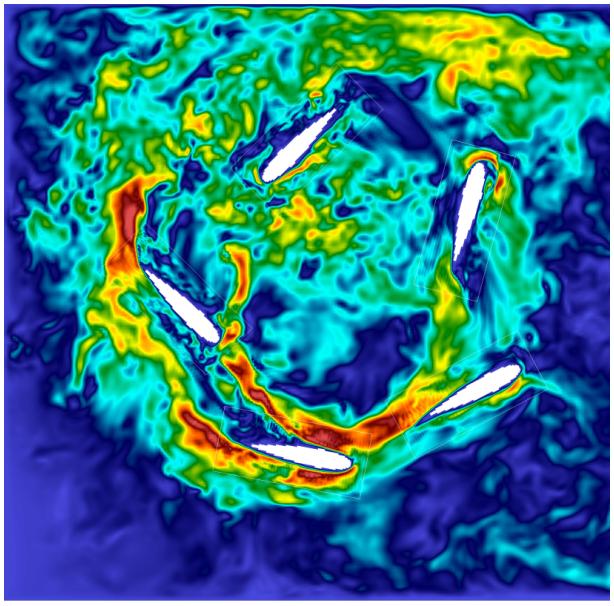


Figure 7.10: VSP velocity field with overset grids.

overset patches. A phenomenon already described in section 5.6. Referring to other cases though we can assume that a further grid refinement will decrease the severity thereof.

The presented case again highlights the improvements gained from overset grids. It is furthermore shown that cases with multiple overset patches moving independently can be computed with ELBE with reasonable effort.

Chapter 8

Conclusion

As part of the presented work a novel approach for moving overset grids for the Lattice Boltzmann Method was developed, implemented and tested within the lattice Boltzmann code ELBE. The main motivation was the improvement of the accuracy of simulations of flows around moving bodies. Starting from the Navier-Stokes equations in moving reference frames transformation formulations were developed in order to allow for the transfer of particle distributions functions from inertial to arbitrarily oriented non-inertial reference frames. Interface tracking algorithms were developed that identify the donor and receptor nodes of two coupled grids. The algorithms were specifically designed to allow for an extensive parallelization on GPUs using the CUDA parallel computing platform. The developed method was succesfully verified and validated against two- and three-dimensional benchmark cases. In the following we shall discuss the main conclusions drawn from the presented work and give a brief outlook of possible future improvements to the method.

8.1 Review of Methods

The direct comparison of the developed approach to the conventional method for moving geometries in ELBE reveals that the addressed accuracy issues can be overcome by applying moving overset grids. The overset approach, widely used within Navier-Stokes FV or FD solvers, should thus also be considered as a beneficial alternative for moving body representations in the LBM.

The application to benchmark cases and comparison to the standard approach has shown that even cases where no obvious numerical degeneracies can be observed with the standard moving body approach overset cases yield generally more accurate results. Moreover it should be stressed that the presented method only utilises the most simple approaches available for two crucial subroutines, namely the forcing term by Luo (1993) and the linear interpolation. Further improvements with respect to accuracy can therefore be expected when applying more complex formulations.

Furthermore, the application of overset grids in ELBE demonstrates that the tracking of interface nodes, generally crucial for the computational performance of overset procedures, is suitable for a massive parallelization on GPUs. This again is, of course, benefiting from the equidistant Cartesian nature of the utilised grids in this LBM approach.

8.2 Future Work

The application and testing of the developed procedure revealed several aspects that should be addressed by future work. The order of appearance in the following outline shall also refer the importance of the respective aspect for the overall performance of the method.

The Forcing Term

The verification study clearly shows that serious errors can be induced by the utilised forcing term. This generally applies to situations where the forcing term is not constant in space and non-parallel to the velocity field. The latter can be concluded both from the presented numerical experiments and the mathematical formulation of the error term. The use of this forcing formulation in this work is historically reasoned. We can generally expect significant accuracy improvements when applying other forcing terms where the additional error vanishes with respect to the LBE without forcing. After all, the implementation thereof is relatively straightforward.

Complex Interpolation

Linear interpolations are the most simple formulation to be applied to transfer particle distribution functions from one grid to another. Even though computationally efficient and easy to implemented, with respect to the LBM they lack one order of accuracy when it comes to the momentum. With the complex interpolation by Geier et al. (2009a) this issue can by overcome while still only requiring the direct neighbour nodes of the receptor node. As the method is generally complex to implement a prior thorough analysis of the error induced by the linear interpolation in actual applications should be considered. Note that for the cases shown in this work a clear distinction between errors caused by the forcing and the interpolation could not be made.

Cross-interface Boundary Conditions

The current overset interface treatment does not cover the case of solid nodes residing on either side of the interface. As shown for the Potsdam Propeller Test Case such situations are generally likely to occur, especially for technical applications with rotating machine parts. In such a case the interpolation to one of the neighbouring fluid nodes on the connected grids consequently lacks a donor node. To capture this case the bounce-back schemes require an additional communication across the interface. Also, it can be desired to place a moving grid with one face along the boundary of the background domain. PDFs could thus be prescribed directly from a boundary condition instead of the background grid. Both adaptations generally do not comprise any new numerical method. They simply require the inclusion of the developed transformation algorithms to further subroutines.

Analysis of Computational Performance

For GPU-accelerated LBM codes like ELBE computational performance is of particular importance. Due to the limitations of this work the developed method was not tested with respect to computational performance. Here, the reference performance would be

given by the standard moving body approach. A detailed analysis measuring the run time of all subroutines is thus suggested. Besides performance improvements of each specific algorithm it is generally recommendable to apply further simple improvements like parallelizing certain kernels of the connectors using CUDA streams.

Bibliography

Andereck, C. D., Liu, S. S., and Swinney, H. L. (1986). Flow regimes in a circular Couette system with independently rotating cylinders. *Journal of Fluid Mechanics*, 164:155–183.

Angerbauer, R. (2015). *Numerische Simulation von Propellerumströmungen mit einem GPU-beschleunigten Gitter-Boltzmann-Verfahren*. Research project, Hamburg University of Technology.

Arney, D. and Flaherty, J. E. (1990). An Adaptive Mesh-moving and Local Refinement Method for Time-dependent Partial Differential Equations. *ACM Trans. Math. Softw.*, 16(1):48–71.

Atta, E. (1981). Component-Adaptive Grid Interfacing. *Proceedings of the AlAA 19th Aerospace and Sciences Meeting, January 12–15, 1981, St.Louis, Missouri*.

Barkmann, U. (2011). Potsdam Propeller Test Case (PPTC) - Open Water Tests with the Model Propeller VP1304. Technical Report Report 3752, SVA Potsdam Model Basin.

Basso, E. and Azevedo, J. L. F. (2004). Three-Dimensional Viscous Flow Simulations over the VLS Using Overset Grids. *J. of the Braz. Soc. of Mech. Sci. & Eng*, 14(4):438–445.

Benek, J. A. and Steger, J. L.and Dougherty, F. C. (1983). A flexible grid embedding technique with application to the Euler equations. *Proceedings of the 6th Computational Fluid Dynamics Conference, July 13-15, 1983, Danvers, MA, USA*.

Bhatnagar, P., Gross, E., and Krook, M. (1954). A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys. Rev.*, 94:511–525.

Bouzidi, M., Firdaouss, M., and Lallemand, P. (2001). Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13(11):3452–3459.

Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 1(1):25–30.

Buick, J. M. and Greated, C. A. (2000). Gravity in a lattice Boltzmann model. *Phys. Rev. E*, 61:5307–5320.

Caiazzo, A. (2008). Analysis of lattice Boltzmann nodes initialisation in moving boundary problems. *Progress in Computational Fluid Dynamics, an International Journal*, 8(1-4):3–10.

Celeritas Simulation Technology (2011). *Overset Grid Assembly Process*. Available from: www.celeritassimtech.com. [Accessed: 10th September 2016].

Chen, H., Filippova, O., Hoch, J., Molvig, K., Shock, R., Teixeira, C., and Zhang, R. (2006). Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Physica A: Statistical Mechanics and its Applications*, 362(1):158 – 167.

Chen, H. and Orszag, S. (2011). Moment isotropy and discrete rotational symmetry of two-dimensional lattice vectors. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 369(1944):2176–2183.

Dellar, P. J. (2002). Nonhydrodynamic modes and *a priori* construction of shallow water lattice Boltzmann equations. *Phys. Rev. E*, 65:036309.

Dellar, P. J. (2003). Incompressible limits of lattice boltzmann equations using multiple relaxation times. *J. Comput. Phys*, page 2003.

d'Humières, D., Ginzburg, I., Krafczyk, M., Lallemand, P., and Luo, L.-S. (2002). Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society A*, 360(1792):437–451.

Donea, J., Huerta, A., Ponthot, J.-P., and Rodriguez-Ferran, A. (2004). *Encyclopedia of Computational Mechanics*, chapter 14: Arbitrary Lagrangian-Eulerian Methods. London, UK: John Wiley & Son.

Exa Corporation (2013). *PowerFLOW Product Brief [Online]*. Available from: http://exa.com/product/simulation-tools/powerflow-cfd-simulation. [Accessed: 14th September 2016].

Filippova, O. and Hänel, D. (1998). Grid Refinement for Lattice-BGK Models. *Journal of Computational Physics*, 147(1):219 – 228.

Flusser, J., Zitova, B., and Suk, T. (2009). *Moments and Moment Invariants in Pattern Recognition*. Chichester, UK: John Wiley and Sons.

Fork, W. and Jürgens, B. (2002). *Faszination Voith- Schneider- Propeller. Geschichte und Technik*. Hamburg, Germany: Koehlers Verlagsges.

Frisch, U., Hasslacher, B., and Pomeau, Y. (1986). Lattice-Gas Automata for the Navier-Stokes Equation. *Phys. Rev. Lett.*, 56:1505–1508.

Gehrke, M. (2015). Weiterentwickung und Validierung eines GPU-beschleunigten, nichtuniformen Gitter-Boltzmann-Verfahrens für turbulente Strömungen. Master's thesis, Technische Universität Hamburg-Harburg.

Geier, M., Greiner, A., and Korvink, G. J. (2009a). Bubble functions for the lattice Boltzmann method and their application to grid refinement. *The European Physical Journal Special Topics*, 171(1):173–179.

Geier, M., Greiner, A., and Korvink, J. (2006). Cascaded digital lattice Boltzmann automata for high Reynolds number flow. *Phys. Rev. E*, 73:066705.

Geier, M., Greiner, A., and Korvink, J. (2009b). A factorized central moment lattice Boltzmann method. *Eur. Phys. J. Special Topics*, 171:55–61.

Geier, M., Schönherr, M., Pasquali, A., and Krafczyk, M. (2015). The cumulant lattice Boltzmann equation in three dimensions: Theory and validation. *Computers & Mathematics with Applications*, 70(4):507–547.

Geller, S. (2010). *Ein explizites Modell für die Fluid-Struktur-Interaktion basierend auf LBM und p-FEM*. PhD thesis, Technischen Universität Carolo-Wilhelmina zu Braunschweig.

Ginzburg, I., Verhaeghe, F., and d'Humières, D. (2008). Two-Relaxation-Time Lattice Boltzmann Scheme: About Parametrization, Velocity, Pressure and Mixed Boundary Conditions. *Communications in Computational Physics*, 3(2):427–478.

Guo, Z., Zheng, C., and Shi, B. (2002a). Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys. Rev. E*, 65:046308.

Guo, Z., Zheng, C.-G., and B.-C., S. (2002b). Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method. *Chinese Physics*, 11(4):366.

Hadžić, H. (2006). *Development and Application of a Finite Volume Method for the Computation of Flows Around Moving Bodies on Unstructured, Overlapping Grids*. PhD thesis, Hamburg University of Technology.

He, X. and Luo, L.-S. (1997). Lattice boltzmann model for the incompressible navier–stokes equation. *Journal of Statistical Physics*, 88(3):927–944.

Henshaw, W. D. (1994). A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids. *Journal of Computational Physics*, 113(1):13–25.

Janßen, C. F. (2010). *Kinetic approaches for the simulation of non-linear free surface flow problems in civil and environmental engineering*. PhD thesis, Technischen Universität Carolo-Wilhelmina zu Braunschweig.

Janßen, C. F., Mierke, D., Überrück, M., Gralher, S., and Rung, T. (2015). Validation of the GPU-Accelerated CFD Solver ELBE for Free Surface Flow Problems in Civil and Environmental Engineering. *Computation*, 3(3):354.

Junk, M. and Yang, Z. (2009a). Convergence of lattice Boltzmann methods for Navier–Stokes flows in periodic and bounded domains. *Numerische Mathematik*, 112(1):65–87.

Junk, M. and Yang, Z. (2009b). Pressure boundary condition for the lattice Boltzmann method. *Computers & Mathematics with Applications*, 58(5):922–929.

Körner, C., Thies, M., Hofmann, T., Thürey, N., and Rüde, U. (2005). Lattice Boltzmann Model for Free Surface Flow for Modeling Foaming. *Journal of Statistical Physics*, 121(1):179–196.

Krafczyk, M. (2001). *Gitter-Boltzmann-Methoden: Von der Theorie zur Anwendung*. Habilitation dissertaion, Technische Universität München.

Lagrava, D., Malaspinas, O., Latt, J., and Chopard, B. (2012). Advances in multi-domain lattice Boltzmann grid refinement. *Journal of Computational Physics*, 231(14):4808 – 4822.

Lallemand, P. and Luo, L.-S. (2000). Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E*, 61:6546–6562.

Lallemand, P. and Luo, L.-S. (2003). Lattice Boltzmann method for moving boundaries. *Journal of Computational Physics*, 184(2):406–421.

Li, L., Sherwin, S. J., and Bearman, P. W. (2002). A moving frame of reference algorithm for fluid/structure interaction of rotating and translating bodies. *International Journal for Numerical Methods in Fluids*, 38(2):187–206.

Li, Y. (2011). *An improved volumetric LBM boundary approach and its extension for sliding mesh simulation*. PhD thesis, Iowa State University.

Lun, C., Savage, S. B., Jeffrey, D. J., and Chepurniy, N. (1984). Kinetic theories for granular flow: inelastic particles in Couette flow and slightly inelastic particles in a general flowfield. *Journal of Fluid Mechanics*, 140:223–256.

Luo, L.-S. (1993). *Lattice-Gas Automata and Lattice Boltzmann Equation for Two- Dimensional Hydrodynamics*. PhD thesis, Georgia Institute of Technology.

McNamara, G. and Zanetti, G. (1988). Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. *Phys. Rev. Lett.*, 61:2332–2335.

Mei, R., Luo, L.-S., Lallemand, P., and d'Humières, D. (2006). Consistent initial conditions for lattice Boltzmann simulations. *Computers & Fluids*, 35(8âĂŞ9):855 – 862. Proceedings of the First International Conference for Mesoscopic Methods in Engineering and Science.

Mei, R., Luo, L.-S., and Shyy, W. (1999). An Accurate Curved Boundary Treatment in the Lattice Boltzmann Method. *Journal of Computational Physics*, 155(2):307 – 330.

Mierke, D. (2015). Ein effizientes und genaues Simulationsverfahren zur Modellierung der gekoppelten Fluid-Eis-Schiff-Interaktion auf Grafikkarten. Master's thesis, Hamburg University of Technology.

Mierke, D., Janßen, C. F., and Rung, T. (2015). GPU-accelerated large-eddy simulation of ship-ice interactions. *Proceedings of the 6th International Conference on Computational Methods in Marine Engineering, 15–17 June 2015, Rome, Italy*.

Miller, S., Campbell, R., Elsworth, C., Pitt, J., and Boger, D. (2014). An Overset Grid Method for Fluid-Structure Interaction. *World Journal of Mechanics*, 4:217–237.

Mittal, R. and Iaccarino, G. (2005). Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261.

Mittal, S. and Kumar, V. (1999). Finite element study of vortex-induced cross-flow and in-line oscillations of a circular cylinder at low Reynolds numbers. *International Journal for Numerical Methods in Fluids*, 31(7):1087–1120.

Mohammad, A. (2011). *Lattice Boltzmann Method – Fundamentals and Engineering Applications with Computer Codes*. London, UK: Springer.

Nakahashi, K., Togashi, F., and Sharov, D. (2000). Intergrid-Boundary Definition Method for Overset Unstructured Grid Approach. *AIAA Journal*, 38(11):2077–2084.

Nguyen, N.-Q. and Ladd, A. J. C. (2005). Sedimentation of hard-sphere suspensions at low reynolds number. *Journal of Fluid Mechanics*, 525:73–104.

Nishida, H. and Meichin, Y. (2012). Seamless Immersed Boundary Lattice Boltzmann Method for Incompressible Flow Simulation. *Proceedings of the seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 9–13, 2012*.

Noack, R. W. (2005). SUGGAR: a General Capability for Moving Body Overset Grid Assembly. *Proceedings of the 17th AIAA Computational Fluid Dynamics Conference, June 6–9 2006, Toronto, Ontario, Canada*.

Nordsiek, F., Huisman, S. G., van der Veen, R., Sun, C., Lohse, D., and Lathrop, D. (2015). Azimuthal velocity profiles in Rayleigh-stable Taylor-Couette flow and implied axial angular momentum transport. *Journal of Fluid Mechanics*, 774:342–362.

NVIDIA (2015). *CUB Documentation [Online]*. Available from: nvlabs.github.io/cub. [Accessed: 25th September 2016].

Pandya, J., Frink, N., and Noack, R. W. (2005). Progress Toward Overset-Grid Moving Body Capability for USM3D Unstructured Flow Solver. *Proceedings of the 17th AIAA Computational Fluid Dynamics Conference, 6–9 June 2005, Toronto, Ontario Canada*.

Peng, Y. and Luo, L.-S. (2008). A comparative study of immersed-boundary and interpolated bounce-back methods in LBE. *Progress in Computational Fluid Dynamics*, 8(1–4):156–167.

Qian, Y.-H., D'Humières, D., and Lallemand, P. (1992). Lattice BGK models for Navier-Stokes equation. *Europhysics Letters*, 17:479.

Rohde, M., Derksen, J., and Van den Akker, H. (2008). An applicability study of advanced lattice-Boltzmann techniques for moving, no-slip boundaries and local grid refinement. *Computers & Fluids*, 37:1238–1252.

Rubinstein, R. and Luo, L.-S. (2008). Theory of the lattice Boltzmann equation: Symmetry properties of discrete velocity sets. *Phys. Rev. E*, 77:036709.

Sideroff, C. (2015). *Applied CCM Announces Partnership with Celeritas Simulation Technology [Online]*. Available from: www.appliedccm.com/applied-ccm-partners-with-celeritas/. [Accessed: 10th September 2016].

Skillen, A. (2011). *The overset grid method, applied to the solution of the incompressible Navier-Stokes equations in two and three spatial dimensions*. PhD thesis, University of Manchester.

Soong, T. T. (2004). *Fundamentals of Probability and Statistics for Engineers*. Chichester, UK: John Wiley and Sons.

Succi, S. (2001). *The Lattice Botzmann Equation for Fluid Dynamics and beyond*. Oxford, UK: Clarendon Press.

Sukop, M. and Thorne, D. (2007). *Lattice Boltzmann Modeling*. Berlin, Germany: Springer.

Tang, H., Casey Jones, S., and Sotiropoulos, F. (2003). An overset-grid method for 3D unsteady incompressible flows. *Journal of Computational Physics*, 191(2):567–600.

Thompson, J., Soni, B., and Weatherill, N., editors (1999). *Handbook of Grid Generation*. Boca Raton, USA: CRC Press.

Tölke, S., Freudiger, S., and Krafczyk, M. (2006). An adaptive scheme using hierarchical grids for lattice Boltzmann multi-phase flow simulations. *Computers & Fluids*, 35:820–830.

Valero-Lara, P. and Jansson, J. (2015). Multi-domain Grid Refinement for Lattice-Boltzmann Simulations on Heterogeneous Platforms. *Proceedings of the 2015 IEEE 18th International Conference on Computational Science and Engineering (CSE)*, pages 1–8.

Wiki2 (2016). *Cyclorotor [Online]*. Available from: en.wiki2.org/wiki/Voith_Schneider_Propeller. [Accessed: 12th October 2016].

Wolfram, S. (1986). Cellular automaton fluids 1: Basic theory. *Journal of Statistical Physics*, 45(3):471–526.

Wood, S. and Deiterding, R. (2015). A lattice Boltzmann method for horizontal axis wind turbine simulation. *Proceedings of the 14th International Conference on Wind Engineering, Porto Alegre, Brazil, June 21-26, 2015*.

Yu, Z. and Fan, L.-S. (2009). An interaction potential based lattice Boltzmann method with adaptive mesh refinement (AMR) for two-phase flow simulation. *Journal of Computational Physics*, 228(17):6456–6478.

Zhang, R., Sun, C., Li, Y., Satti, R., Shock, R., Hoch, J., and Chen, H. (2015). Lattice Boltzmann Approach for Local Reference Frames. *Communications in Computational Physics*, 9(5):1193–1205.

Zou, Q. and He, X. (1997). On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids*, 9(6):1591–1598.