Combining computational fluid dynamics and magnetic resonance imaging data using lattice Boltzmann based topology optimisation

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

von der KIT-Fakultät für Mathematik des
Karlsruher Instituts für Technologie (KIT)
genehmigte

DISSERTATION

von
Dipl.-Math. Fabian Klemens
aus Kandel

Tag der mündlichen Prüfung: 30. September 2020

Referentin: PD Dr. Gudrun Thäter
Korreferent: Prof. Dr. Willy Dörfler
Korreferent: Dr. Mathias J. Krause

Parts of this thesis have been published in advance:

F. Klemens, S. Schuhmann, G. Guthausen, G. Thäter, M. J. Krause. *CFD-MRI: a coupled measurement and simulation approach for accurate fluid flow characterisation and domain identification.* Computers & Fluids, 2018 [1]

F. Klemens, B. Förster, M. Dorn, G. Thäter, M. J. Krause. *Solving fluid flow domain identification problems with adjoint lattice Boltzmann methods.* Computers & Mathematics with Applications, 2020 [2]

F. Klemens, S. Schuhmann, R. Balbierer, G. Guthausen, H. Nirschl, G. Thäter, M. J. Krause. *Noise reduction of flow MRI measurements using a lattice Boltzmann based topology optimisation approach.* Computers & Fluids, 2020 [3]

Combining computational fluid dynamics and magnetic resonance imaging data using lattice Boltzmann based topology optimisation

This thesis presents the combination of magnetic resonance imaging (MRI) measurements and computational fluid dynamics (CFD) to reduce statistical measurement noise and identify objects and finer structures in the MRI data. Using a lattice Boltzmann based topology optimisation approach, the method allows those solutions that best match the measured flow field but satisfy the macroscopic conservation laws of fluid flow, here mass and momentum conservation. This combination is formulated as a distributed control problem that minimises the distance between measured and simulated flow field, the latter being the solution of a parametrised Boltzmann equation with Bhatnagar–Gross–Krook collision operator, where the controls represent the porosity distributed in the domain. The problem is solved with an adjoint lattice Boltzmann method using the open source software OpenLB.

This thesis was developed in the Lattice Boltzmann Research Group (LBRG) at the Institute for Applied and Numerical Mathematics (IANM) and the Institute for Mechanical Process Engineering and Mechanics (MVM) of the Karlsruhe Institute of Technology (KIT).

The research leading to these results has received funding from the Federal Ministry of Education and Research (BMBF), the German Research Foundation (DFG), and the Heidelberg Academy of Sciences and Humanities (HAW) for the projects *"Lehre hoch Forschung: Problemorientiertes, interdisziplinäres Softwarepraktikum Mathematik"*, *"Charakterisierung von durchströmten Gebieten und Strömungen mittels modell- und simulationsbasierter Fluss-MRI (CFD-MRI)"*, and *"Charakterisierung von durchströmten Gefäßen und der Hämodynamik mittels modell- und simulationsbasierter Fluss-MRI (CFD-MRI)"*.

Most computations were performed on the supercomputer ForHLR funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

Contents

Chapter 1

Introduction

Magnetic resonance imaging (MRI), in particular phase contrast MRI, which is capable of imaging fluid flows in complex geometries, is a very versatile and important tool not only in medical but also in technical applications. In medical applications it can help to identify severe health problems such as heart disease much more accurately, e.g. by locating arterial stenoses or by identifying plaque in the arteries. In technical applications, the identification of problems in filtration processes is of great importance, for example in water treatment, in order to detect the accumulation of organic and inorganic substances in filters, which lead to productivity losses and increasing maintenance and operating costs.

These measurements, however, are corrupted by noise, especially in time-critical imaging. Therefore, in time-dependent processes, e.g. filtration processes, where the development of membrane fouling is of importance, or the imaging of patients, where time is certainly a limiting factor, noise becomes a major challenge for the analysis of the data. In addition, there is no explicit method to determine other properties such as wall shear stresses or pressure gradients as they are not directly observable by MRI.

Computational fluid dynamics (CFD) is a highly valuable addition and even alternative to measurements, as it is not time-sensitive, cheap and can provide a lot of additional information. It is also very easy and inexpensive to track and monitor the effects of changes, like the influence of different shapes on drag in aerodynamics. However, flow simulations are always only an image of reality, a model in which many simplifications have to be made. In the medical and technical applications mentioned above, this includes above all the true inner geometries and structures of the observed objects.

Therefore, a combination of measurement and simulation promises a great benefit, which can combine the strengths of both methods to depict complex, realistic flow processes, to

be free of noise and make a multitude of analyses possible.

The combination of observations and model is called data assimilation [4]. A prominent example of data assimilation is numerical weather forecasting [5], where observations such as e.g. temperature and pressure measurements are fed into the model to improve the forecast. The coupling of flow models, such as the Navier–Stokes equations, and MRI data is mainly used in medical applications [6–8]. Data assimilation is used here to support e.g. finding aneurysms in $4D$ flow MRI data [9]. However, these methods are mostly limited to boundary control problems where the underlying inner and outer geometry is already known, mostly by computer tomography (CT) imaging. But taking the information from the MRI to build a simulation can be difficult or impossible, for example if the internal structure of the flow domain is not known. This is certainly the case with noisy images of a filter, where the outer geometry is known, e.g. a pipe flow, but the geometry of the complex accumulation on the filter is not.

In this thesis a method is presented which combines simulation and measurement using a topology optimisation approach, called CFD-MRI [1]. Here objects and geometries of the MRI measurement are found automatically [2], with simultaneous noise reduction [3].

Topology optimisation for fluid flows was first presented by Borrvall and Petersson [10] to minimise the power dissipation in a Stokes flow. This was extended to incompressible Navier–Stokes problems by Gersborg-Hansen [11]. Both were using finite element methods to discretise the problem. For lattice Boltzmann methods the first to use topology optimisation were Pingen et al. [12]. Since then the methods have been improved [13–16] and the area of application has been widely extended, for example minimising pressure in non-Newtonian flow [15], in transient flow [17–20], or improving qualities of multiphase [21–23] and thermal flows [24, 25]. The first to use adjoint lattice Boltzmann methods were Tekitek et al. [26] who solved parameter identification problems. They used the first-discretise-then-optimise procedure [27, 28], where the adjoint problem is derived after discretising the problem. The other way, first-optimise-then-discretise, was first proposed by Krause et al. [29]. In 2016 Krause et al. [30] applied topology optimisation to domain identification problems, where a known, simulated velocity field was used for the identification of a cubic object in the flow, which is the basis of this work.

This thesis is structured as follows. The first chapters shall provide a basis for the methods and equations used. The aim is to standardise the terms and methods used and to help to cover all necessary basics within this thesis. First, different approaches for modelling fluid flows and the governing equations are presented in Chapter 2. The chapter concludes with the numerical algorithm to simulate fluid flows as well as its validation by means of numerical experiments of flow in porous media, which is needed to set up the

proposed optimisation problem. For this, an analytical solution for flow in porous media in three dimensions is derived. Chapter 3 then gives a brief introduction to MRI, the physical basics and equations of how to measure flows, as well as the problem of measurement noise and its mathematical description. The latter is used for the validation of the presented method with regard to noise reduction. Since the aim of this thesis is the coupling of CFD and MRI data by means of topology optimisation, Chapter 4 presents the basics of optimisation theory, such as the explanation of the terminology, concepts, existence statements and also the methods used for the numerical computation of optimal control problems. This includes a line search method with Wolfe conditions, which is implemented to improve the stability of the optimisation, and was validated using a simple optimisation problem. The detailed description of the CFD-MRI method takes place in Chapter 5. The idea and implementation are motivated and the necessary equations are derived and discretised. The derivation of the needed adjoint problem is done here with a formal Lagrange technique. A proof of concept for CFD-MRI with real data is shown in Chapter 6. For this, the implementation is extended to use differently available data, that is data with lower dimension, fewer velocity components and arbitrary location. The method is then numerically validated. Chapter 7 analyses the CFD-MRI method to identify objects based only on velocity data in the observed domain. Thereby a simple and a complex object to be identified are considered and the velocity data is gradually reduced to determine the applicability of the method, and binary classification is used for the validation of the object identification. The ability of the method to reduce measurement noise is analysed in Chapter 8. For this, synthetic data is created based on real MRI data in order to allow the evaluation of the CFD-MRI results. Finally, in Chapter 9 the method is applied to data of a flow through a porous structure as they appear in filtration applications. Here, the object identification of the porous structure and noise reduction of the flow data are analysed in combination. Additionally, the implementation is improved with regard to performance by employing the steady state solution and explicit time marching for the adjoint problem. The main contribution of this thesis is therefore the extension, improvement and thorough analysis of the method to use MRI data.

Chapter 2

Fluid dynamics

In this chapter the basics of fluid dynamics are shown. The behaviour of fluids can be described in different ways. Macroscopically, which leads to the famous Navier–Stokes equations, and statistically, which leads to the equally famous Boltzmann equation. A discretisation strategy that combines these two descriptions is the lattice Boltzmann method which discretises a certain form of the Boltzmann equation to approximate solutions to the Navier–Stokes equations. This chapter concludes with a numerical experiment to validate the used model.

2.1 Navier–Stokes equations

The Navier–Stokes equations describe fluid flow by fundamental physical conservation laws, here mass, momentum and energy conservation. Often, the conservation of energy is not explicitly given, and it is also not shown here. For a detailed introduction to the topic the works of Wesseling [31], Feistauer [32], or Ferziger [33] are recommended.

Let $\Omega \subset \mathbb{R}^d$ be the domain with dimension $d \in \{2, 3\}$ and $I = [0, T]$ the time interval with $T > 0$. The most important quantities of hydrodynamics are density $\rho : I \times \Omega \to \mathbb{R}$, velocity $\boldsymbol{u} : I \times \Omega \to \mathbb{R}^d$ and pressure $p : I \times \Omega \to \mathbb{R}$.

For the derivation of the Navier–Stokes equations presented here, two fundamental theorems of are needed. The general Leibniz theorem, also called *Reynolds transport theorem*, and the Gauss, or *divergence theorem*.

Theorem 2.1 (Reynolds transport theorem)**.** *Let $\boldsymbol{\phi}(t, \boldsymbol{x})$ be a sufficient smooth function in a time-varying connected domain of $\mathbb{R}^d$ for $d \in \{2, 3\}$, then it holds*

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{V(t)} \boldsymbol{\phi}(t, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int_{V(t)} \partial_t \boldsymbol{\phi}(t, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} + \int_{\partial V(t)} \boldsymbol{\phi} \boldsymbol{u} \cdot \boldsymbol{n} \, \mathrm{d}s, \tag{2.1}$$

where $\boldsymbol{u}$ is the Eulerian velocity of the boundary ∂V and $\boldsymbol{n}$ its unit outward normal.

Proof. See for example Theorem 1.3.1 in [31], or Theorem 1.5 in [32]. □

Theorem 2.2 (Divergence Theorem)**.** *For any volume $V \subset \mathbb{R}^d$ with piecewise smooth boundary ∂V, it holds*

$$\int_V \nabla \cdot \boldsymbol{\phi} \, \mathrm{d}\boldsymbol{x} = \int_{\partial V} \boldsymbol{\phi} \cdot \boldsymbol{n} \, \mathrm{d}s, \tag{2.2}$$

where $\boldsymbol{\phi}$ is a continuously differentiable vector field and $\boldsymbol{n}$ is the unit outward normal on the boundary ∂V.

Proof. See [31], Theorem 1.2.2. □

2.1.1 Mass conservation

The first property of fluids and a fundamental law of physics is the conservation of mass, where it is assumed that mass is neither created nor destroyed. Mathematically this reads

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_V \rho(t, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = 0, \tag{2.3}$$

where $V \subset \Omega$ is an arbitrary control volume.

Applying Reynolds transport theorem and Gauss divergence theorem, it follows

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \int_V \rho(t, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} &= \int_V \partial_t \rho(t, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} + \int_{\partial V} \boldsymbol{u}(t, \boldsymbol{x}) \rho(t, \boldsymbol{x}) \cdot \boldsymbol{n} \, \mathrm{d}s \\ &= \int_V \Big(\partial_t \rho(t, \boldsymbol{x}) + \nabla \cdot \big(\boldsymbol{u}(t, \boldsymbol{x}) \rho(t, \boldsymbol{x}) \big) \Big) \, \mathrm{d}\boldsymbol{x} \\ &= 0. \end{aligned} \tag{2.4}$$

The control volume V can be chosen arbitrarily and thus

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 \qquad \text{in } I \times \Omega. \tag{2.5}$$

This is called the *continuity equation* and describes the conservation of mass. Although the derivation was technical, its properties have a physical meaning. After rewriting (2.4) it follows

$$\int_V \partial_t \rho(t, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = - \int_{\partial V} \boldsymbol{u}(t, \boldsymbol{x}) \rho(t, \boldsymbol{x}) \cdot \boldsymbol{n} \, \mathrm{d}s, \tag{2.6}$$

which describes that a change of mass in an observed volume can only occur when there is a flux of mass through its boundary.

2.1.2 Momentum conservation

The next important conservation law, which corresponds to Newton's second law of motion [34], is the conservation of momentum. It states that the momentum only changes if there is a force acting and reads

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_V \rho(t, \boldsymbol{x}) \boldsymbol{u}(t, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \boldsymbol{F}(t, \boldsymbol{x}). \tag{2.7}$$

The net force $\boldsymbol{F} : I \times \Omega \to \mathbb{R}^d$ can be split into internal and external forces [35], i.e. $\boldsymbol{F} = \boldsymbol{F}_{\text{external}} + \boldsymbol{F}_{\text{internal}}$. The external forces, such as gravity, buoyancy or electromagnetic forces, act on the volume [33] by

$$\boldsymbol{F}_{\text{external}} = \int_V \rho(t, \boldsymbol{x}) \boldsymbol{f}(t, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \tag{2.8}$$

where $\boldsymbol{f} : I \times \Omega \to \mathbb{R}^d$ describes the body forces per unit mass [36]. The internal, or contact forces, such as pressure or shear stresses, which act on the boundary [33] can be described by

$$\boldsymbol{F}_{\text{internal}} = \int_{\partial V} \boldsymbol{\sigma}(t, \boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) \, \mathrm{d}s, \tag{2.9}$$

where $\boldsymbol{\sigma} : I \times \Omega \to \mathbb{R}^{d \times d}$ is the so-called *stress tensor*.

For the left hand side of (2.7) and in the same way as for the continuity equation it follows

$$\int_V \Big(\partial_t \big(\rho(t, \boldsymbol{x}) \boldsymbol{u}(t, \boldsymbol{x}) \big) + \nabla \cdot \big(\rho(t, \boldsymbol{x}) \boldsymbol{u}(t, \boldsymbol{x}) \otimes \boldsymbol{u}(t, \boldsymbol{x}) \big) \Big) \, \mathrm{d}\boldsymbol{x}, \tag{2.10}$$

where $\otimes : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^{m \times n}$ for $m, n \in \mathbb{N}$ denotes the outer product.

For the right hand side of (2.7) we employ again the divergence theorem and thus

$$\int_{\partial V} \boldsymbol{\sigma}(t,\boldsymbol{x}) \cdot \boldsymbol{n} \, \mathrm{d}s = \int_V \nabla \cdot \boldsymbol{\sigma}(t,\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \tag{2.11}$$

With this, and the again using that V can be chosen arbitrarily, the equation of momentum conservation now reads

$$\partial_t(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{f} \qquad \text{in } I \times \Omega. \tag{2.12}$$

The external forces $\boldsymbol{f}$ are given by the observed problem, e.g. if gravity or coriolis forces are acting, but the exact form and the properties of the stress tensor $\boldsymbol{\sigma}$ are still unknown.

Stress tensor

In order to get an explicit form of the stress tensor, some assumptions have to be made, called Stokes postulates (cf. [32]).

The first assumption is that the stress tensor is composed of the pressure forces, and the shear stresses acting on the fluid, and thus

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{S}, \tag{2.13}$$

where the *shear stress* $\boldsymbol{S} \in \mathbb{R}^{d\times d}$ should be a continuous function f of the *deformation tensor* $\boldsymbol{D}(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^\top)$. Additionally, if the deformation tensor is zero, it is assumed that only pressure acts on the surface.

If the fluid is assumed to be isotropic the shear stress is invariant for transformations in space, i.e.

$$\boldsymbol{A}\boldsymbol{S}\boldsymbol{A}^{-1} = f\big(\boldsymbol{A}\boldsymbol{D}(\boldsymbol{u})\boldsymbol{A}^{-1}\big), \tag{2.14}$$

for every orthonormal matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$.

If additionally the relation of deformation and stress is linear, then the stress tensor can be derived to have following form [32, 37]

$$\boldsymbol{\sigma}(t,\boldsymbol{x}) = -p(t,\boldsymbol{x})\boldsymbol{I} + 2\mu \boldsymbol{D}(\boldsymbol{u})(t,\boldsymbol{x}) - \lambda \nabla \cdot \boldsymbol{u}(t,\boldsymbol{x})\boldsymbol{I}, \tag{2.15}$$

where $\lambda, \mu \in \mathbb{R}$ are material constants, and μ is called *dynamic viscosity*.

With this, for the right hand side of (2.12), the divergence of the stress tensor reads

$$\nabla \cdot \boldsymbol{\sigma}(t, \boldsymbol{x}) = -\nabla p(t, \boldsymbol{x}) + \mu \Delta \boldsymbol{u}(t, \boldsymbol{x}) - (\lambda + \mu) \nabla \big(\nabla \cdot \boldsymbol{u}(t, \boldsymbol{x}) \big), \tag{2.16}$$

and the *Navier–Stokes equations* can be derived as

$$\partial_t(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) - \mu \Delta \boldsymbol{u} + (\lambda + \mu) \nabla (\nabla \cdot \boldsymbol{u}) + \nabla p = \rho \boldsymbol{f} \qquad \text{in } I \times \Omega, \tag{2.17a}$$

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 \qquad \text{in } I \times \Omega. \tag{2.17b}$$

For *incompressible fluids*, i.e. $\partial_t \rho = 0$, it follows $\nabla \cdot \boldsymbol{u} = 0$ for the continuity equation and thus

$$\rho(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) - \mu \Delta \boldsymbol{u} + \nabla p = \rho \boldsymbol{f} \qquad \text{in } I \times \Omega, \tag{2.18a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } I \times \Omega. \tag{2.18b}$$

For (2.18) to be well posed, boundary and initial condition have to be defined, for example $\boldsymbol{u}(0, \cdot) = 0$, i.e. a fluid at rest in the beginning. Here only Dirichlet boundary conditions are considered to define function values on the boundary. These are used for example for constant pressure, given velocity profile at the inflow, or no-slip boundary condition, i.e. $\boldsymbol{u}(\cdot, \boldsymbol{x}) = 0$ ($\boldsymbol{x} \in \partial\Omega$).

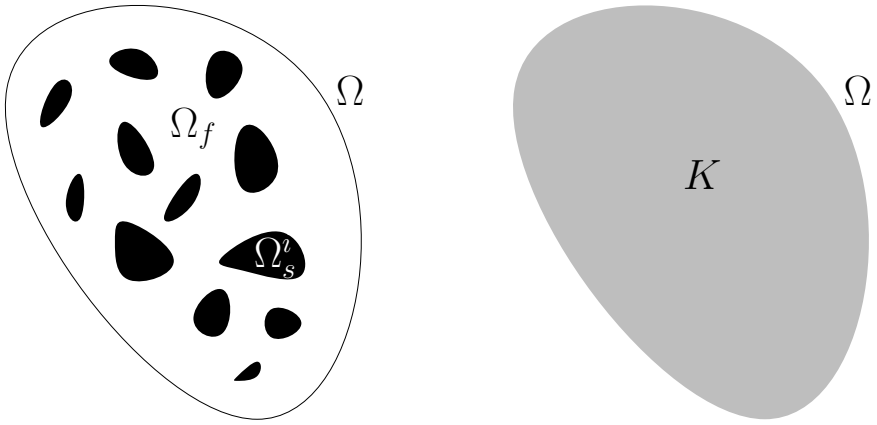
Showing the existence, uniqueness and smoothness of solutions to the Navier–Stokes equations with suitable initial and boundary conditions in three dimensions is still an open problem, and is one of the *Millennium Prize Problems* stated by the Clay Mathematics Institute. A good overview of existing proofs and the official problem statement can be found in [38].

2.1.3 Flow in porous media

Flow in porous media is a very important problem in sciences, engineering and even medicine, for example to investigate groundwater flows, air flow in lungs or filtration processes in water treatment. In this thesis, porous media flow is important for the use of topology optimisation problems to ensure continuous functions, see Chapter 4.

A porous medium can be described, on a very small scale, by a structure with holes, or by solid objects distributed in a domain. On a larger scale, these structures appear as continuous medium with quantities like *porosity*, which is the fraction of fluid to whole volume, and *permeability*, which is a measure for the ability of the medium to transmit

fluid (cf. Figure 2.1).



(a) Small scale (b) Large scale

Figure 2.1: Scheme of a porous medium. Left, the small scale of a porous medium with solid, non-permeable objects distributed in the domain. Right, the large scale of the porous medium with permeability as averaged quantity.

Let's consider a domain Ω, called porous medium, which is composed of $N_s \in \mathbb{N}$ solid, non-permeable objects denoted by Ω_s^i for $i = 0, 1, \ldots, N_s - 1$ and fluid domain Ω_f, see Figure 2.1a. Assuming low velocity and a steady state solution the convective term $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ and the time derivative $\partial_t \boldsymbol{u}$ can be neglected. Additionally, a no-slip boundary condition on the boundaries of the objects, here denoted as $\partial\Omega_s^i$, is assumed and thus this system reads

$$-\mu \Delta \boldsymbol{u} + \nabla p = 0 \qquad \text{in } \Omega_f, \tag{2.19a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega_f, \tag{2.19b}$$

$$\boldsymbol{u} = 0 \qquad \text{on } \partial\Omega_s^i,\ i = 0, 1, \ldots, N_s - 1. \tag{2.19c}$$

In 1856 Darcy [39] did experiments on flow of water through beds of sand and found the following empirical one-dimensional relationship

$$Q = -\frac{K}{\mu} \frac{\triangle P}{L}, \tag{2.20}$$

where Q is the volumetric flow rate, $\triangle P$ the pressure difference along a characteristic length L, and K the permeability.

Many approaches have been developed to derive Darcy's law (2.20) from the Stokes equations (2.19). One of them is called the *method of volume averaging* [40]. For this the

superficial and intrinsic volume averages are defined as

$$\langle \boldsymbol{\phi} \rangle \stackrel{\text{def}}{=} \frac{1}{V} \int_{\Omega_f} \boldsymbol{\phi} \, \mathrm{d}\boldsymbol{x}, \qquad \langle \boldsymbol{\phi} \rangle^f \stackrel{\text{def}}{=} \frac{1}{V_f} \int_{\Omega_f} \boldsymbol{\phi} \, \mathrm{d}\boldsymbol{x}, \tag{2.21}$$

where $V = |\Omega|$ and $V_f = |\Omega_f|$. Thus the relation between superficial and intrinsic average is $\langle \boldsymbol{\phi} \rangle_f = \epsilon \langle \boldsymbol{\phi} \rangle$, i.e. $\epsilon = \frac{V_f}{V}$, which is the porosity of the medium. Using the *spatial averaging theorem* [40, 41]

$$\langle \nabla \cdot \boldsymbol{\phi} \rangle = \nabla \cdot \langle \boldsymbol{\phi} \rangle + \frac{1}{V} \int_{\partial \Omega_s} \boldsymbol{n} \cdot \boldsymbol{\phi} \, \mathrm{d}s, \tag{2.22}$$

Whitaker was able to derive Darcy's law by averaging the Stokes equations as follows

$$\langle \boldsymbol{u} \rangle + \frac{\boldsymbol{K}}{\mu} \nabla \langle p \rangle^f = 0 \qquad \text{in } \Omega, \tag{2.23a}$$

$$\nabla \cdot \langle \boldsymbol{u} \rangle = 0 \qquad \text{in } \Omega. \tag{2.23b}$$

Here $\boldsymbol{K} \in R^{d \times d}$ is the symmetric, positive definite *permeability tensor*. For an in-depth derivation of the above equations, see the work of Whitaker [40, 41].

In another approach, based on homogenisation theory [42, 43], the domain is defined by periodically placed solid objects. Here, the periodicity of the objects is defined by a parameter ϵ and is thus different to the volume averaging method, where the domain is fixed. The idea is that the heterogeneous structure of the porous domain will be homogenised by taking the limit of ϵ to zero. This is done using a two-scale asymptotic expansion

$$\boldsymbol{\phi}_\epsilon(t, \boldsymbol{x}) = \sum_{i=0}^{\infty} \epsilon^i \boldsymbol{\phi}_i \left(t, \boldsymbol{x}, \frac{\boldsymbol{x}}{\epsilon} \right), \tag{2.24}$$

where $\boldsymbol{\phi}_i(t, \boldsymbol{x}, \boldsymbol{y})$ is periodic in $\boldsymbol{y}$ with period $Y \stackrel{\text{def}}{=} (0, 1)^d$. This expansion is used for velocity and pressure, i.e $\boldsymbol{\phi}_i \in \{\boldsymbol{u}, p\}$, in the Stokes or Navier–Stokes equations. After arranging the terms according to their order of ϵ the resulting homogenised equations can be derived. Using this, it is possible to derive Darcy, Stokes and the *Brinkman equations* depending on *critical sizes* of the obstacles with respect to the whole domain. For a detailed description of the method and derivation of the equations see the work of Allaire [44, 45].

The Brinkman equations [46, 47] are thereby intermediate between Stokes equations and

Darcy's law and read

$$-\mu_{\text{eff}}\Delta \boldsymbol{u} + \mu \boldsymbol{K}^{-1}\boldsymbol{u} + \nabla p = 0 \qquad \text{in } \Omega, \tag{2.25a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{2.25b}$$

where μ_{eff} is the effective viscosity, and used to model the transition phase between porous medium and fluid [48], and is often assumed to be equal to the fluid viscosity. If the porous medium is isotropic, the permeability tensor $\boldsymbol{K} \in \mathbb{R}^{d\times d}$ reduces to a scalar $K \in \mathbb{R}_{>0}$ [49]. For the Brinkman equations it holds that for $K \to \infty$ this results in the Stokes equations and for $K \to 0$ in Darcy's law. Borvall and Peterson [10] were the first to use (2.25) for topology optimisation in fluid dynamics.

The nonlinear, instationary Brinkman equations, here called Brinkman-type Navier–Stokes equations, were derived by Mikelić [50] through homogenisation as

$$\rho\big(\partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\big) - \mu_{\text{eff}}\Delta \boldsymbol{u} - \mu K^{-1}\boldsymbol{u} + \Delta p = 0 \qquad \text{in } I\times\Omega, \tag{2.26a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } I\times\Omega. \tag{2.26b}$$

If the fluid velocity is high, additionally to the linear Darcy term, $\mu K^{-1}\boldsymbol{u}$, a quadratic Forchheimer term

$$F_e = c_F K^{-\frac{1}{2}}\rho|\boldsymbol{u}|\boldsymbol{u}, \tag{2.27}$$

is added to (2.26) [51]. Ergun [52] did experiments with packed beds of spheres where the following relations were found [53]

$$c_F = \frac{1.75}{\sqrt{150\phi^3}}, \qquad K = \frac{\phi^3 d_p^2}{150(1-\phi)^2}. \tag{2.28}$$

Here, $\phi \in [0,1]$ is the porosity, and $d_p \in \mathbb{R}_{>0}$ the diameter of the spheres. For an overview of models the work of Nield [49] is recommended.

2.2 Boltzmann equation

Since fluids are composed of a vast number of atoms and molecules, it is clear that the Navier–Stokes equations can not paint the whole picture. On the other hand, if considering every interaction of every molecule in some volume of fluid by basic Newton laws, the computation time of even a cubic centimetre of water would be infeasible [54]. The Boltzmann equation is much more general as the Navier–Stokes equations as it combines

the microscopic behaviour of fluids in a statistical manner. By this, the Boltzmann equation is valid not only for continuum mechanics but also for free molecular flow [55].

This section is based on the work of Babovsky [56], Cercignani [54], Hänel [55], Saint-Raymond [57] and Villani [58].

The basic quantity of the Boltzmann equation is the nonnegative distribution function $f(t, \boldsymbol{x}, \boldsymbol{\xi})$, which gives the mass density of particles, e.g. atoms or molecules, with velocity $\boldsymbol{\xi} \in \Xi \subset \mathbb{R}^d$ at time $t \in I = [0, T]$ for $T > 0$ and position $\boldsymbol{x} \in \Omega \subset \mathbb{R}^d$ with dimension $d \in \{2, 3\}$. We further define the *phase space* as $\Omega \times \Xi$.

Per definition one can derive, at least formally, macroscopic quantities like density ρ_f or velocity $\boldsymbol{u}_f$ by integrating the distribution function over the microscopic velocity space, i.e.

$$\rho_f = \int_\Xi f(t, \boldsymbol{x}, \boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}, \qquad \rho_f \boldsymbol{u}_f = \int_\Xi \boldsymbol{\xi} f(t, \boldsymbol{x}, \boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}. \tag{2.29}$$

These quantities are called moments M_ϕ and can be summarised by using

$$M_\phi(t, \boldsymbol{x}) = \int_\Xi \phi(\boldsymbol{\xi}) f(t, \boldsymbol{x}, \boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}, \tag{2.30}$$

where $\phi(\boldsymbol{\xi})$ is some continuous function of the microscopic particle velocity. Higher order moments can be used to derive energy and temperature [56, 59].

The evolution of the distribution function f is described by the Boltzmann equation, which is a balance equation for transportation and collision of microscopic particles in phase space

$$\frac{\mathrm{d}f}{\mathrm{d}t} = Q(f, f) \qquad \text{in } I \times \Omega \times \Xi. \tag{2.31}$$

The left hand side of (2.31), the transport term, considers the change of particles in some control volume of the phase space, due to time and velocity, which can be written as

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \partial_t f + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f. \tag{2.32}$$

This also means that $f(t, \boldsymbol{x}, \boldsymbol{\xi}) = f(0, \boldsymbol{x} - t\boldsymbol{\xi}, \boldsymbol{\xi})$ is a weak solution to

$$\partial_t f + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f = 0, \tag{2.33}$$

i.e. free transport [58], which will be a convenient property for the discretisation done in Section 2.3. Note, if a macroscopic force $\boldsymbol{F}$ is acting on the particles with mass m, an

additional term of $\frac{\mathbf{F}}{m} \cdot \nabla_{\boldsymbol{\xi}} f$ is added to (2.32).

The right hand side of (2.31), the collision term, considers particles which are gained and lost in the control volume, due to collisions. To derive the complex collision term, some assumptions have to be made [54, 55, 57, 58]. Let's assume that the collisions are *binary*, i.e. the collision of more than two particles can be neglected, and the collision happens exactly at $(t, \boldsymbol{x}) \in I \times \Omega$, i.e. the collision time is small compared to the time scale. Further, it is assumed that the collisions are *elastic*, meaning the momentum and energy are conserved, i.e.

$$\boldsymbol{\xi}' + \boldsymbol{\xi}'_* = \boldsymbol{\xi} + \boldsymbol{\xi}_*, \qquad |\boldsymbol{\xi}'|^2 + |\boldsymbol{\xi}'_*|^2 = |\boldsymbol{\xi}|^2 + |\boldsymbol{\xi}_*|^2, \tag{2.34}$$

where $\boldsymbol{\xi}, \boldsymbol{\xi}_*$ and $\boldsymbol{\xi}', \boldsymbol{\xi}'_*$ are the velocities before and after collisions, respectively. These collisions are assumed to be time-reversible and the particles that collide are assumed to be uncorrelated. These assumptions are not further discussed here, but the interested reader is referred to the works of Cercignani [54] and Villani [58].

Using above assumptions, the collision term can be derived as [57]

$$Q(f, f)(t, \boldsymbol{x}, \boldsymbol{\xi}) = \int_{S^{d-1}} \int_{\mathbb{R}^d} k(|\boldsymbol{\xi} - \boldsymbol{\xi}_*|, \boldsymbol{s})(f' f'_* - f f_*) \, \mathrm{d}\boldsymbol{\xi}_* \, \mathrm{d}\boldsymbol{s}, \tag{2.35}$$

where $f \overset{\text{def}}{=} f(t, \boldsymbol{x}, \boldsymbol{\xi})$, $f' \overset{\text{def}}{=} f(t, \boldsymbol{x}, \boldsymbol{\xi}')$, $f_* \overset{\text{def}}{=} f(t, \boldsymbol{x}, \boldsymbol{\xi}_*)$, $f'_* \overset{\text{def}}{=} f(t, \boldsymbol{x}, \boldsymbol{\xi}'_*)$. Additionally, $k(|\boldsymbol{\xi} - \boldsymbol{\xi}_*|, \boldsymbol{s})$ is defined as the *collision kernel*, where $\boldsymbol{s}$ is the unit vector on the unit sphere S^{d-1} in which the collisions are observed.

Summarised, the Boltzmann equation reads

$$\partial_t f + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f = \int_{S^{d-1}} \int_{\Xi} k(|\boldsymbol{\xi} - \boldsymbol{\xi}_*|, \boldsymbol{s})(f' f'_* - f f_*) \, \mathrm{d}\boldsymbol{\xi}_* \, \mathrm{d}\boldsymbol{s}. \tag{2.36}$$

2.2.1 Equilibrium distribution function

The Boltzmann equation (2.36) is a complex, nonlinear integro-differential equation, but it is possible to derive interesting properties. One of them are the existence of *collision invariants*.

Definition 2.1 (Collision invariant)**.** *A local integrable function $\phi : \mathbb{R}^d \to \mathbb{R}$ is a* collision invariant *if for all $f \in L^1(\mathbb{R}^d)$ with integrable ϕf, it holds*

$$\int_{\mathbb{R}^d} \phi(\boldsymbol{\xi}) Q(f, f) \, \mathrm{d}\boldsymbol{\xi} = 0. \tag{2.37}$$

For these collision invariant, the following can be derived.

Lemma 2.1. *For a collision invariant ϕ it follows*

$$\phi(\boldsymbol{\xi}) + \phi(\boldsymbol{\xi}_*) = \phi(\boldsymbol{\xi}') + \phi(\boldsymbol{\xi}'_*). \tag{2.38}$$

Proof. This can be shown by using the symmetric structure of the collision operator and interchanging the role of $\boldsymbol{\xi}$ and $\boldsymbol{\xi}_*$, see e.g. Lemma 2.18 in [56]. □

As it can be directly seen, the moments of density and momentum (2.29), are collision invariants. Also, one implication of Lemma 2.1 is that every collision invariant can be expressed by $\phi(\boldsymbol{\xi}) = a + \boldsymbol{b} \cdot \boldsymbol{\xi} + c|\boldsymbol{\xi}|^2$, for $a, c \in \mathbb{R}$ and $\boldsymbol{b} \in \mathbb{R}^d$, see e.g. [56].

Definition 2.2 (Equilibrium distribution function)**.** *A distribution function $f > 0$ is called equilibrium distribution function, if*

$$Q(f, f) = 0. \tag{2.39}$$

Taking a look at the collision term (2.35), it can be seen that

$$Q(f, f) = 0 \Leftrightarrow f' f'_* = f f_* \text{ a.e.}, \tag{2.40}$$

and thus

$$\ln f' + \ln f'_* = \ln f + \ln f_*. \tag{2.41}$$

Therefore it follows from Lemma 2.1 that $\ln(f)$ is a collision invariant if $\ln(f)Q(f, f) \in L^1(\Xi)$. With this, and using the assumptions of the elastic collision in (2.34) and the conservation of mass it follows that the equilibrium distribution function can be derived as

$$f^{eq}_{\rho,\boldsymbol{u},T}(t, \boldsymbol{x}, \boldsymbol{\xi}) = \frac{\rho(t, \boldsymbol{x})}{(2\pi RT)^{d/2}} \exp\left(-\frac{(\boldsymbol{\xi} - \boldsymbol{u}(t, \boldsymbol{x}))^2}{2RT}\right), \tag{2.42}$$

where T is the absolute temperature, and R the universal gas constant, which can be derived from the energy conservation [55, 56].

H theorem

A very important feature of the Boltzmann equation is that it satisfies the second law of thermodynamics. For this Boltzmann's H functional is defined as

$$H(f) \stackrel{\text{def}}{=} \iint_{\Omega\times\Xi} f \ln(f)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{\xi}, \tag{2.43}$$

where f solves the Boltzmann equation. The H functional can be linked to the entropy which is stated in the famous H theorem of Boltzmann (cf. [58]).

Theorem 2.3 (H theorem)**.** *For the H functional* (2.43) *it holds*

$$\frac{\mathrm{d}}{\mathrm{d}t}H\big(f(t,\cdot,\cdot)\big) = \int_\Omega\int_\Xi \big(Q(f,f)\ln(f)\big)(t,\boldsymbol{x},\cdot)\,\mathrm{d}\boldsymbol{\xi}\,\mathrm{d}\boldsymbol{x} \le 0, \tag{2.44}$$

i.e. it is decreasing in time. And $\frac{\mathrm{d}}{\mathrm{d}t}H = 0$ *if and only if* $f = f^{eq}$*.*

Proof. See for example [60, p. 137ff.]. □

One consequence of the H theorem is the intrinsic irreversibility, which connects the Boltzmann equation to continuum mechanics [58]. This irreversibility is especially interesting since reversibility is one of the assumptions to derive the Boltzmann equation [54].

BGK collision operator

Bhatnagar, Gross and Krook [61] introduced a collision operator that is considerably simpler than the original one, while retaining the important properties, including the validity of the H theorem and the conservation of macroscopic quantities. The *BGK collision operator* $J(f)$ is essentially a relaxation of the distribution function to the equilibrium and reads

$$J(f) \stackrel{\text{def}}{=} -\frac{1}{\tau}(f - f^{eq}), \tag{2.45}$$

where τ is the *relaxation time*, which is related to the transport coefficients, like the viscosity of the underlying fluid [59].

With this, the BGK-Boltzmann equation reads

$$\partial_t f + \boldsymbol{\xi}\cdot\nabla_{\boldsymbol{x}} f = -\frac{1}{\tau}(f - f^{eq}) \qquad \text{in } I\times\Omega\times\Xi. \tag{2.46}$$

2.2.2 Conservation laws

Let f be a solution of the Boltzmann equation (2.31), then it follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Xi}\phi(\boldsymbol{\xi})f(t,\boldsymbol{x},\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}=\int_{\Xi}\phi(\boldsymbol{\xi})Q(f,f)(t,\boldsymbol{x},\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}. \tag{2.47}$$

Using the Reynolds transport theorem 2.1 for the left hand side, and if $\phi(\boldsymbol{\xi})$ is a collision invariant, it follows

$$\partial_t\int_{\Xi}\phi(\boldsymbol{\xi})f(t,\boldsymbol{x},\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}+\nabla\cdot\int_{\Xi}\boldsymbol{\xi}\phi(\boldsymbol{\xi})f(t,\boldsymbol{x},\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}=0. \tag{2.48}$$

Thus, for the collision invariant $\phi\equiv 1$ one gets

$$\partial_t\rho_f+\nabla\cdot(\rho_f\boldsymbol{u}_f)=0, \tag{2.49}$$

which is the continuity equation (2.5) from Navier–Stokes equations. For $\phi\equiv\boldsymbol{\xi}$ it follows, at least formally (cf. [57, 58]),

$$\partial_t(\rho_f\boldsymbol{u}_f)+\nabla\cdot\big(\rho_f(\boldsymbol{u}_f\otimes\boldsymbol{u}_f)+\boldsymbol{\sigma}\big)=0. \tag{2.50}$$

This is the conservation of momentum as in (2.12), and like for the Navier–Stokes equations, the exact form of the stress tensor $\boldsymbol{\sigma}$ is not known [55].

In order to find the exact form of the stress tensor, and to derive the (incompressible) Navier–Stokes equations (2.18) directly from the Boltzmann equation (2.36), the *Chapman-Enskog expansion* [62] has become an established method [54, 58, 59].

The fundamental idea of the Chapman-Enskog expansion is the perturbation of the distribution function for small deviations to the equilibrium, written as

$$f_\epsilon=f^{\mathrm{eq}}+\sum_{i=1}^{N}\epsilon^i f^{(i)}, \tag{2.51}$$

where ϵ is related to the *Knudsen number*, which describes the ratio of microscopic to macroscopic length scales, and $f^{(i)}$ for $i=1,\dots,N$ is the i-th order perturbation of the equilibrium. This expansion is then used in the Boltzmann equation and solved for the different orders N of the perturbation function [55, 56, 59]. This procedure is similar to the asymptotic expansion in homogenisation theory of porous media done by Allaire, see Section 2.1.3. Note that although the Chapman-Enskog expansion is a widely used procedure, it is not rigorously mathematically justified, see for example [58]. Nonetheless it is an important tool and for example used to derive an explicit form of the stress

tensor, see Section 2.3. A more rigorous way to derive Navier–Stokes equations from the BGK-Boltzmann equation was shown for example by Saint-Raymond [63].

2.2.3 Boundary conditions

Again, for the system to be complete, boundary conditions need to be defined. For this, let $\partial\Omega$ be the smooth boundary of the stationary spatial domain $\Omega \subset \mathbb{R}^d$, with $\boldsymbol{n}$ the unit outward normal on $\partial\Omega$.

One boundary condition, called *specular reflection*, assumes that microscopic particles hitting the boundary rebound at the same angle, like billiard balls hitting a wall. Mathematically, this reads

$$f(\cdot, \boldsymbol{x}, \boldsymbol{\xi}) = f(\cdot, \boldsymbol{x}, \boldsymbol{\xi}_n) \qquad \text{on } \partial\Omega, \tag{2.52}$$

where $\boldsymbol{\xi}_n = \boldsymbol{\xi} - 2\,(\boldsymbol{\xi} \cdot \boldsymbol{n})\,\boldsymbol{n}$. Although this is a very natural condition, as it assumes a perfectly smooth boundary this can lead to problems as particle interactions are not taken into account, see e.g. [58].

Another boundary condition, called *bounce back*, assumes that particles which collide with the boundary will reverse their velocity, and thus

$$f(\cdot, \boldsymbol{x}, \boldsymbol{\xi}) = f(\cdot, \boldsymbol{x}, -\boldsymbol{\xi}) \qquad \text{on } \partial\Omega. \tag{2.53}$$

In contrast to the specular reflection boundary condition, this does not seem natural. But it has some very useful properties, namely the no-slip condition, when regarding the discretised version, see Section 2.3.2.

For more informations about boundary conditions for the Boltzmann equation, see for example Cercignani [54], or Villani [58].

2.2.4 Existence of solutions

The global existence of solutions of the BGK-Boltzmann equation have been shown for example by Perthame in [64] and [65]. For a more general overview on solutions to the Boltzmann equation the work of Cercignani [60] is recommended.

Let's consider the BGK-Boltzmann equation with Dirichlet-type boundary conditions in a bounded domain

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f + f - f^{\mathrm{eq}}_{\rho,\boldsymbol{u},T} = 0 \qquad \text{in } I \times \Omega \times \Xi, \tag{2.54a}$$

$$f(t, \boldsymbol{x}, \boldsymbol{x}) = \phi(\boldsymbol{x}, \boldsymbol{\xi}) \qquad \text{on } I \times \partial\Omega \times \Xi, \tag{2.54b}$$

$$f(0, \boldsymbol{x}, \boldsymbol{\xi}) = f_0(\boldsymbol{x}, \boldsymbol{\xi}) \qquad \text{in } \Omega \times \Xi, \tag{2.54c}$$

with equilibrium distribution function

$$f^{\mathrm{eq}}_{\rho,\boldsymbol{u},T}(t, \boldsymbol{x}, \boldsymbol{\xi}) = \frac{\rho(t, \boldsymbol{x})}{\big(2\pi T(t, \boldsymbol{x})\big)^{3/2}} \exp\left(-\frac{|\boldsymbol{\xi} - \boldsymbol{u}(t, \boldsymbol{x})|^2}{2T(t, \boldsymbol{x})}\right). \tag{2.55}$$

For (2.54) the following theorem holds

Theorem 2.4 (Existence of solutions of the BGK-Boltzmann equation)**.** *Let* $\phi(\boldsymbol{\xi}) \in L^1(\Xi)$ *with* $|\boldsymbol{\xi}|^2 \phi(\boldsymbol{x}, \boldsymbol{\xi}) \leq \phi(\boldsymbol{\xi})$ *for all* $\boldsymbol{x} \in \partial\Omega$ *and* $\boldsymbol{\xi} \in \Xi$*, and*

$$\int_{\Omega\times\Xi} f_0(\boldsymbol{x}, \boldsymbol{\xi}) \left(1 + |\boldsymbol{\xi}|^2 + |\log f_0|\right) \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{\xi} < +\infty, \qquad f_0(\boldsymbol{x}, \boldsymbol{\xi}) \geq 0, \tag{2.56}$$

$$\int_{\partial\Omega\times\Xi} \phi(\boldsymbol{x}, \boldsymbol{\xi}) |\boldsymbol{\xi} \cdot \boldsymbol{n}| \left(1 + |\boldsymbol{\xi}|^2 + |\log \phi|\right) \mathrm{d}s\, \mathrm{d}\boldsymbol{\xi} < +\infty, \qquad \phi(\boldsymbol{x}, \boldsymbol{\xi}) \geq 0, \tag{2.57}$$

then there exist a solution to (2.54)*.*

Proof. See the work of Perthame [65]. □

2.3 Lattice Boltzmann equation

The lattice Boltzmann method (LBM) is a numerical method which leads to efficient and fast computational algorithms for the simulation of fluid dynamics. In contrast to conventional methods, such as finite volume or finite element, the discretisation is based on the BGK-Boltzmann equation and not on the classical Navier–Stokes or advection-diffusion equations. But since the Boltzmann equation is more general, it is possible to solve various fluid problems. This includes multiphase flows [59, 66, 67], thermal flows [59, 67–69], flow in porous media [66, 67, 70–72], turbulent flows [67, 73, 74], fluid structure interactions [59], particulate flows [75–77], sound waves [59, 78], and even light simulation [79, 80].

For a comprehensive overview of the lattice Boltzmann method, the interested reader is

referred to the work of Wolf-Gladrow [81], Sukop and Thorne [66], Guo and Shu [67], and especially the work of Krüger et al. [59].

2.3.1 Discretisation

The discretisation of the BGK-Boltzmann equation in order to approximate the Navier–Stokes equations is done here in three main steps. First, Taylor expansion of the equilibrium distribution function in order to use numerical integration. Second, Gauss–Hermite quadrature of the moments, resulting in a velocity-discrete Boltzmann equation. Third, forward Euler for the discretisation of time and space, leading to the lattice Boltzmann equation.

Starting from the equilibrium distribution function, where here only the dependency on the velocity is denoted, it follows

$$\begin{aligned} f^{\mathrm{eq}}(\boldsymbol{\xi}) &= \frac{\rho}{(2\pi RT)^{d/2}} \exp\left(\frac{-(\boldsymbol{\xi}-\boldsymbol{u})^2}{2RT}\right) \\ &= \frac{\rho}{(2\pi RT)^{d/2}} \exp\left(\frac{-\boldsymbol{\xi}^2}{2RT}\right) \exp\left(\frac{\boldsymbol{\xi}\cdot\boldsymbol{u}}{RT} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2RT}\right) \\ &= \omega(\boldsymbol{\xi})\rho \exp\left(\frac{\boldsymbol{\xi}\cdot\boldsymbol{u}}{RT} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2RT}\right), \end{aligned} \tag{2.58}$$

with $\omega(\boldsymbol{\xi}) \stackrel{\mathrm{def}}{=} \frac{1}{(2\pi RT)^{d/2}} \exp\left(\frac{-\boldsymbol{\xi}^2}{2RT}\right)$. Using Taylor expansion for the second term, considering $\frac{|\boldsymbol{u}|}{\sqrt{RT}} \ll 1$, also known as low Mach number expansion [82], this leads to

$$f^{\mathrm{eq}}(\boldsymbol{\xi}) = \underbrace{\omega(\boldsymbol{\xi})\rho\left(1 + \frac{\boldsymbol{\xi}\cdot\boldsymbol{u}}{RT} + \frac{(\boldsymbol{\xi}\cdot\boldsymbol{u})^2}{2(RT)^2} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2RT}\right)}_{\stackrel{\mathrm{def}}{=} \tilde{f}^{\mathrm{eq}}(\boldsymbol{\xi})} + \mathcal{O}(\boldsymbol{u}^3). \tag{2.59}$$

Using Hermite polynomials the same as above can be derived, with two additional properties. First, using the Hermite polynomials for the expansion does not need the low Mach number assumption, and second it can be shown that the first three moments, i.e. density, velocity and energy are conserved, as they are directly linked to the coefficients of the orthonormal Hermite basis [83].

Velocity-discrete Boltzmann equation

Discretising the velocity space it must be ensured that following integrals, the moments, hold

$$\int_{\Xi} f^{\mathrm{eq}} \,\mathrm{d}\boldsymbol{\xi} = \rho, \qquad \int_{\Xi} \boldsymbol{\xi} f^{\mathrm{eq}} \,\mathrm{d}\boldsymbol{\xi} = \rho \boldsymbol{u}. \tag{2.60}$$

A closer look at (2.59) reveals a structure that can be numerically integrated by Gauss–Hermite quadrature

Theorem 2.5 (Gaussian quadrature)**.** *Let $p \in \mathcal{P}_N$, i.e. a polynomial function of degree N, then*

$$\int_{\Omega} \omega(\boldsymbol{x}) p(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \sum_{i=0}^{q-1} w_i p(\boldsymbol{x}_i), \tag{2.61}$$

where $q \geq 2N+1$ and $\boldsymbol{x} \in \Omega \subseteq \mathbb{R}^d$. The weights w_i and the abscissae $\boldsymbol{x}_i$ are found as the zeroes of orthogonal polynomials $p_n(\boldsymbol{x})$ corresponding to the weight function $\omega(\boldsymbol{x})$. For $\omega(\boldsymbol{x}) = e^{-\boldsymbol{x}^2}$ the orthogonal polynomials are the Hermite polynomials $\mathcal{H}_n(\boldsymbol{x})$.

Proof. See Davis and Rabinowitz [84]. □

In order to conserve the hydrodynamic moments, following must therefore hold

$$\sum_{i=0}^{q-1} \phi(\boldsymbol{\xi}_i) f^{\mathrm{eq}}(\boldsymbol{\xi}_i) \overset{!}{=} \int_{\Xi} \phi(\boldsymbol{\xi}) f^{\mathrm{eq}}(\boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi} = \begin{cases} \rho, & \phi \equiv 1 \\ \rho \boldsymbol{u}, & \phi \equiv \boldsymbol{\xi} \end{cases}. \tag{2.62}$$

Let $p(\boldsymbol{\xi})$ be a polynomial defined as

$$p(\boldsymbol{\xi}) = \phi(\boldsymbol{\xi}) \left(1 + \frac{\boldsymbol{\xi} \cdot \boldsymbol{u}}{RT} + \frac{(\boldsymbol{\xi} \cdot \boldsymbol{u})^2}{2(RT)^2} - \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2RT} \right). \tag{2.63}$$

Further, let $\hat{\boldsymbol{\xi}} \overset{\mathrm{def}}{=} \frac{\boldsymbol{\xi}}{\sqrt{2RT}}$, $q(\hat{\boldsymbol{\xi}}) \overset{\mathrm{def}}{=} p(\sqrt{2RT}\boldsymbol{\xi})$ and $w_i \overset{\mathrm{def}}{=} \frac{\hat{w}_i}{(\pi)^{d/2}}$. Then, using the Gauss–

Hermite quadrature, it follows

$$\begin{aligned}\int \phi(\boldsymbol{\xi})\tilde{f}^{\mathrm{eq}}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} &= \int \frac{\rho}{(2\pi RT)^{d/2}} \exp\left(\frac{-\boldsymbol{\xi}^2}{2RT}\right) p(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} \\ &= \frac{\rho}{(\pi)^{d/2}} \int \exp\left(-\hat{\boldsymbol{\xi}}^2\right) q(\hat{\boldsymbol{\xi}})\,\mathrm{d}\hat{\boldsymbol{\xi}} \\ &= \frac{\rho}{(\pi)^{d/2}} \sum_{i=0}^{q-1} \hat{w}_i q(\hat{\boldsymbol{\xi}}_i) \\ &= \sum_{i=0}^{q-1} \phi(\boldsymbol{\xi}_i) \underbrace{w_i \rho \left(1 + \frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{RT} + \frac{(\boldsymbol{\xi}_i \cdot \boldsymbol{u})^2}{2(RT)^2} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2RT}\right)}_{\stackrel{\mathrm{def}}{=} f^{\mathrm{eq}}(t,\boldsymbol{x},\boldsymbol{\xi}_i)}.\end{aligned} \tag{2.64}$$

Additional requirements on the weights w_i can be formulated, for example as $\sum_{i=0}^{q-1} f_i^{\mathrm{eq}} \stackrel{!}{=} \rho$, it follows

$$\sum_{i=0}^{q-1} w_i \stackrel{!}{=} 1, \qquad \sum_{i=0}^{q-1} w_i \boldsymbol{\xi}_i \stackrel{!}{=} 0, \qquad \sum_{i=0}^{q-1} w_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i \stackrel{!}{=} (RT)^2 \boldsymbol{I}. \tag{2.65}$$

Depending on the dimension d and the number of discrete velocities q the resulting set of weights and abscissae is commonly denoted as $DdQq$. For $D3Q19$, which is used in this thesis, the weights w_i are

$$w_i = \begin{cases} \frac{1}{3}, & i = 0 \\ \frac{1}{18}, & i = 1,\ldots,6 \\ \frac{1}{36}, & i = 7,\ldots,18 \end{cases}, \tag{2.66}$$

and the abscissae, i.e. the discrete velocities $\boldsymbol{\xi}_i$, are

$$\boldsymbol{\xi}_i = \begin{cases} \big(0,0,0\big)^\top & \text{for } i = 0 \\ \big(\pm\sqrt{3},0,0\big)^\top, \big(0,\pm\sqrt{3},0\big)^\top, \big(0,0,\pm\sqrt{3}\big)^\top & \text{for } i = 1,\ldots,6 \\ \big(\pm\sqrt{3},\pm\sqrt{3},0\big)^\top, \big(\pm\sqrt{3},0,\pm\sqrt{3}\big)^\top, \big(0,\pm\sqrt{3},\pm\sqrt{3}\big)^\top & \text{for } i = 7,\ldots,18 \end{cases}. \tag{2.67}$$

For a closer look at the derivation and further $DdQq$ sets the work of Shan et al. [83] is recommended. Note that for three dimensions a quadrature rule with twenty-seven abscissae, i.e. $D3Q27$, would be needed to integrate the moments exactly, but using symmetry properties the number of abscissae can be reduced to nineteen, i.e. $D3Q19$ [59, 83].

Defining $\boldsymbol{c}_i \stackrel{\mathrm{def}}{=} \frac{\boldsymbol{\xi}_i}{\sqrt{3}}$ and using (2.65) it follows that $RT = \frac{1}{3}$. Further, $RT \stackrel{\mathrm{def}}{=} c_s^2$, where

c_s can later be related to the speed of sound. With this the velocity-discrete equilibrium distribution function can be written as

$$f_i^{\mathrm{eq}}(t,\boldsymbol{x}) = w_i\rho(t,\boldsymbol{x})\left(1+\frac{\boldsymbol{c}_i\cdot\boldsymbol{u}(t,\boldsymbol{x})}{c_s^2}+\frac{\big(\boldsymbol{c}_i\cdot\boldsymbol{u}(t,\boldsymbol{x})\big)^2}{2c_s^4}-\frac{\boldsymbol{u}(t,\boldsymbol{x})\cdot\boldsymbol{u}(t,\boldsymbol{x})}{2c_s^2}\right), \quad (2.68)$$

with $f_i^{\mathrm{eq}}(t,\boldsymbol{x}) \stackrel{\mathrm{def}}{=} f^{\mathrm{eq}}(t,\boldsymbol{x},\boldsymbol{c}_i)$, and discrete velocity space $Q \stackrel{\mathrm{def}}{=} \{\boldsymbol{c}_i : i=0,\dots,q-1\} \subset \Xi$.

The discretisation of the velocity space $\Xi \subset \mathbb{R}^d$ leads to the velocity-discrete BGK-Boltzmann equation

$$\frac{\partial f_i}{\partial t} + \boldsymbol{c}_i\cdot\nabla_{\boldsymbol{x}} f_i = -\frac{1}{\tau}(f_i - f_i^{\mathrm{eq}}) \qquad \text{in } I\times\Omega, \quad (2.69)$$

for $i=0,\dots,q-1$, and $f_i(t,\boldsymbol{x}) \stackrel{\mathrm{def}}{=} \frac{w_i}{\omega(\boldsymbol{c}_i)} f(t,\boldsymbol{x},\boldsymbol{c}_i)$ [59]. The weights w_i and the abscissae $\boldsymbol{c}_i$ are thereby depending for example on the dimension and the order of the Gauss–Hermite quadrature [82, 83]. With this, the moments can be computed by (cf. [59])

$$\rho(t,\boldsymbol{x}) = \sum_{i=0}^{q-1} f_i^{\mathrm{eq}}(t,\boldsymbol{x}) \quad = \sum_{i=0}^{q-1} f_i(t,\boldsymbol{x}), \quad (2.70)$$

$$\rho\boldsymbol{u}(t,\boldsymbol{x}) = \sum_{i=0}^{q-1} \boldsymbol{c}_i f_i^{\mathrm{eq}}(t,\boldsymbol{x}) = \sum_{i=0}^{q-1} \boldsymbol{c}_i f_i(t,\boldsymbol{x}). \quad (2.71)$$

Lattice Boltzmann equation

In order to obtain the fully discrete lattice Boltzmann equation, the time $I=[0,T],\ T>0$ and spatial domain $\Omega\subset\mathbb{R}^d$ need to be discretised.

Considering the velocity-discrete BGK-Boltzmann equation and rewriting the transport term as in (2.32), it follows for $i=0,\dots,q-1$,

$$\frac{\mathrm{d}f_i}{\mathrm{d}t} + \frac{1}{\tau}f_i = \frac{1}{\tau}f_i^{\mathrm{eq}} \qquad \text{in } I\times\Omega. \quad (2.72)$$

Recalling that $f(t,\boldsymbol{x},\boldsymbol{\xi}) = f(0,\boldsymbol{x}-t\boldsymbol{\xi},\boldsymbol{\xi})$ is a weak solution to the free transport equation (2.33) (cf. [55, 58]) and using the method of characteristics [59], (2.72) is transformed into an ordinary differential equation which can be solved, resulting in (cf. [59, 82])

$$f_i(t+\triangle t,\boldsymbol{x}+\boldsymbol{c}_i\triangle t) = e^{-\frac{\triangle t}{\tau}}\left(f_i(t,\boldsymbol{x}) + \frac{1}{\tau}\int_t^{t+\triangle t} e^{-\frac{t'-t}{\tau}} f_i^{\mathrm{eq}}\big(t',\boldsymbol{x}+\boldsymbol{c}_i(t'-t)\big)\,\mathrm{d}t'\right). \quad (2.73)$$

Using forward Euler for the remaining integral and Taylor expansion for the exponential

functions the *lattice Boltzmann equation* (LBE) can be derived as [59, 82]

$$\underbrace{f_i(t+\triangle t, \boldsymbol{x}+\boldsymbol{c}_i\triangle t)}_{\text{stream}} = \underbrace{f_i(t,\boldsymbol{x}) - \frac{\triangle t}{\tau}\big(f_i(t,\boldsymbol{x}) - f_i^{\text{eq}}(t,\boldsymbol{x})\big)}_{\text{collide}}. \tag{2.74}$$

It can be shown that by using Crank–Nicolson for the integral term and second order expansion of the exponential functions the exact same equation can be derived and thus make the lattice Boltzmann method formally a second order method [85].

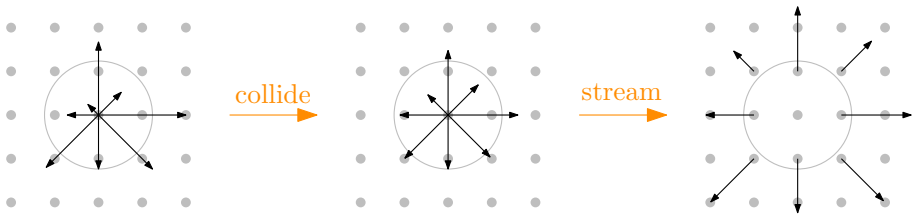


Figure 2.2: Illustration of the collide and stream steps of the lattice Boltzmann equation. The collision step is purely local, and the streaming step only depends on the direct neighbours. This locality leads to highly parallelisable algorithms.

The discrete time space can now be defined as $I_h \overset{\text{def}}{=} \{t \in I : t = t_0 + n\triangle t,\, n \in \mathbb{N}\} \subset I$, and due to the discrete velocity space Q it follows that the discrete spatial domain $\Omega_h \subset \Omega$ is a regular grid, called lattice, where for $\boldsymbol{x} \in \Omega_h$ it follows that $\boldsymbol{x} + \boldsymbol{c}_i\triangle t \in \Omega_h$.

The lattice Boltzmann equation (2.74) is conceptually parted into two steps, a local collision step (*collide*) and a subsequent streaming step (*stream*), which only depends on the direct neighbouring nodes. This locality is the reason for the high parallelisable structure of the LBE, making the lattice Boltzmann method an efficient and computationally fast method [86, 87].

Using Chapman–Enskog analysis it is possible to obtain the Navier–Stokes equations through (2.74) when following relations hold [55, 59, 67]

$$\mu = \rho c_s^2\left(\tau - \frac{\triangle t}{2}\right), \qquad p = \rho c_s^2. \tag{2.75}$$

Thus, the viscosity μ is related to the relaxation time τ, and c_s can now be identified as the speed of sound [55, 59]. The modelling error thereby is of second order in the Mach number, which results in accuracy of first order in time and second in space [59].

2.3.2 Initial and boundary conditions

One major difference for initial and boundary conditions in the lattice Boltzmann context is that the macroscopic quantities are defined by the distribution functions. While in other methods the macroscopic quantities can be set directly, here schemes are needed to convert these quantities in values of the distribution functions.

One example for initial conditions is the definition of distribution functions by their local equilibrium, i.e.

$$f_i(0,\cdot) = f_i^{\mathrm{eq}}(0,\cdot), \tag{2.76}$$

where $f_i^{\mathrm{eq}} = w_i\rho\left(1 + \frac{\boldsymbol{c}_i\cdot\boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{c}_i\cdot\boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2c_s^2}\right)$ for $i = 0,\ldots,q-1$ and with given density ρ and velocity $\boldsymbol{u}$. These are commonly set to $\rho(0,\cdot) = 1$ and $\boldsymbol{u}(0,\cdot) = \mathbf{0}$ [59].

There are a variety of different schemes for the application of boundary conditions, the most prominent being the so-called bounce back condition.

Bounce back

As already stated in Section 2.2.3 the bounce back condition can, in continuous setting, be written as $f(\cdot,\cdot,\boldsymbol{\xi}) = f(\cdot,\cdot,-\boldsymbol{\xi})$, which holds on the boundary. In discrete setting this condition will lead to the *no-slip condition*, i.e. zero velocity at the boundary.

The bounce back rule in discrete setting is thereby defined as (cf. [88])

$$f_{i'}(t+\triangle t, \boldsymbol{x}_f) = f_i^c(t, \boldsymbol{x}_f), \tag{2.77}$$

where $\boldsymbol{x}_s = \boldsymbol{x}_f + \boldsymbol{c}_i\triangle t$, i.e. the solid node $\boldsymbol{x}_s$ is a direct neighbour of the fluid node $\boldsymbol{x}_f$ in direction $\boldsymbol{c}_i$ for some $i \in \{0,\ldots,q-1\}$, and i' the index of opposite direction of i, thus $\boldsymbol{c}_{i'} = -\boldsymbol{c}_i$. Also, f_i^c denotes the distribution function after collision. An illustration of the scheme for $D2Q19$ is shown in Figure 2.3.

Because $\boldsymbol{c}_{i'} = -\boldsymbol{c}_i$ and by the bounce back rule it follows

$$\sum_{i=0}^{q-1} \boldsymbol{c}_i f_i(t,\boldsymbol{x}) = \mathbf{0}, \tag{2.78}$$

and thus $\boldsymbol{u}(\cdot,\boldsymbol{x}) = \mathbf{0}$ on the boundary $\partial\Omega_h$.

The bounce back rule is also the foundation of more sophisticated boundary conditions,

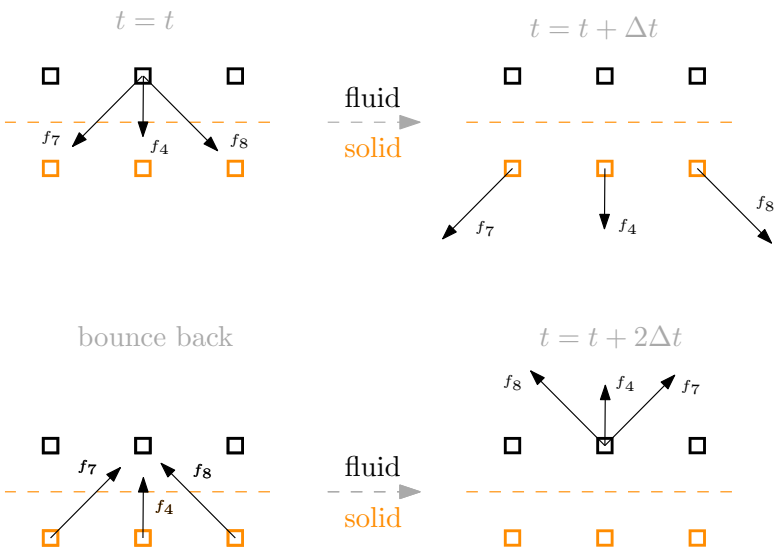


Figure 2.3: Illustration of the bounce back rule with $D2Q9$ model. At time $t = 0$ the populations are shown after the collision step, they are then streamed to the neighbouring nodes. If the nodes are solid the velocities are reversed and the distributions are streamed back to the originating node at time $t = t + 2\triangle t$. This illustration is based on the original figure of Sukop and Or [89].

from which two are presented in the following.

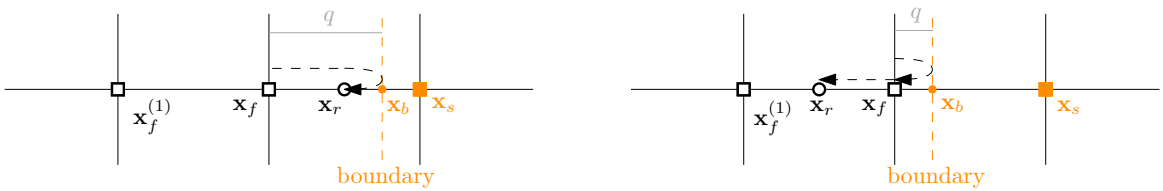
Interpolated bounce back

Because the grid in lattice Boltzmann methods is uniformly spaced, curved boundaries can only be approximated by a so-called stair case when using bounce back which limits the accuracy of the simulations [90]. In order to take into account the exact position of curved boundaries Bouzidi et al. [91] have proposed an interpolation scheme for the bounce back rule which has second order accuracy.

Let $\boldsymbol{x}_f \in \Omega_h$ be a fluid node, $\boldsymbol{x}_s \in \Omega_h$ a solid node, and $\boldsymbol{x}_b$ the exact position of the boundary, which is does not need to be in Ω_h. The distance between the boundary and the nearest fluid node is denoted as

$$q = \frac{|\boldsymbol{x}_b - \boldsymbol{x}_f|}{\triangle \boldsymbol{x}}, \tag{2.79}$$

where $\triangle \boldsymbol{x}$ is the spacing between nodes. Since the distribution functions will always travel exactly a distance of $\boldsymbol{c}_i \triangle t$, taking the exact position of the wall into account will lead to particle distributions which are not defined on the grid after colliding with the boundary, see Figure 2.4. This reflected position is denotes as $\boldsymbol{x}_r$.



(a) $q \geq 1/2$ (b) $q < 1/2$

Figure 2.4: Illustration of the two cases of the interpolated bounce back scheme. Here $\boldsymbol{x}_f$ and $\boldsymbol{x}_f^{(1)}$ denote fluid nodes and $\boldsymbol{x}_s$ a solid node. Distribution functions which are streamed from $\boldsymbol{x}_f$ and get reflected at the exact position of the boundary $\boldsymbol{x}_b$ will travel to a node $\boldsymbol{x}_r$ which is not on the grid. The interpolated bounce back scheme then uses interpolation to give values for the distribution functions at $\boldsymbol{x}_f$. This illustration is based on the original figure of Bouzidi et al. [91].

In order to get the value of the distribution function at $\boldsymbol{x}_r$, Bouzidi et al. used an interpolation scheme with the information of the distribution functions at the nearest fluid node $\boldsymbol{x}_f$ and its direct neighbour $\boldsymbol{x}_f^{(1)}$, together with the bounce back rule (2.77). This leads to following scheme, which depends on the the distance q (cf. [91])

$$f_{i'}(t+\triangle t, \boldsymbol{x}_f) = \begin{cases} 2q f_i^c(t, \boldsymbol{x}_f) + (1-2q) f_i^c(t, \boldsymbol{x}_f^{(1)}) & q < \frac{1}{2} \\ \frac{1}{2q} f_i^c(t, \boldsymbol{x}_f) + \frac{2q-1}{2q} f_{i'}^c(t, \boldsymbol{x}_f) & q \geq \frac{1}{2} \end{cases}. \tag{2.80}$$

Non-equilibrium bounce back

In order to use Dirichlet boundary conditions Zou and He [92] proposed a scheme which has third-order accuracy [59]. Here, the scheme for velocity boundary conditions is shown, but the method can also be used for pressure boundary conditions by using the ideal gas law in LBM, i.e. $p = c_s^2 \rho$.

Let the distribution functions at the boundary, $f_i(\cdot, \boldsymbol{x}_b)$, be unknown for some $i \in \{0, \ldots, q-1\}$. The number of unknown distribution functions thereby depend on the number of discrete velocities, for example with $D2Q9$ three distribution would be unknown, as shown in Figure 2.5.

For Zou–He boundary conditions the first step is to compute the density at the boundary by rearranging the moments, i.e.

$$\sum_{i=0}^{q-1} f_i(\cdot, \boldsymbol{x}_b) = \rho(\cdot, \boldsymbol{x}_b), \qquad \sum_{i=0}^{q-1} \boldsymbol{c}_i f_i(\cdot, \boldsymbol{x}_b) = \rho \boldsymbol{u}(\cdot, \boldsymbol{x}_b), \tag{2.81}$$

according to the unknown distributions and solve for the density. If the density is known,

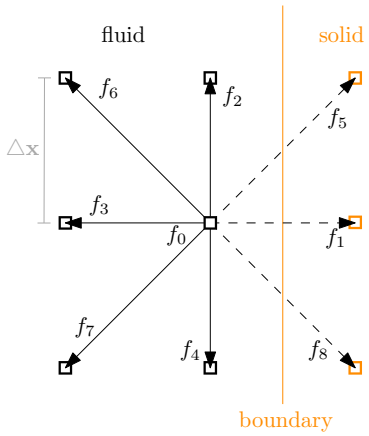


Figure 2.5: Illustration of unknown distribution functions for boundary conditions. For $D2Q9$ and a boundary on the right side, three distribution functions are unknown, here f_1, f_5, and f_8.

then every equilibrium distribution function $f_i^{\mathrm{eq}}(\cdot, \boldsymbol{x}_b)$ is known, as the velocity is given as boundary condition. Using the non-equilibrium bounce back rule [59, 92]

$$f_i(\cdot, \boldsymbol{x}_b) - f_i^{\mathrm{eq}}(\cdot, \boldsymbol{x}_b) = f_{i'}(\cdot, \boldsymbol{x}_b) - f_{i'}^{\mathrm{eq}}(\cdot, \boldsymbol{x}_b), \tag{2.82}$$

where again $\boldsymbol{c}_i = -\boldsymbol{c}_{i'}$, the system can then be solved for the unknown distribution functions.

2.3.3 Porous media flow

As shown in Section 2.1.3, the flow in porous media can be described by Brinkman-type Navier–Stokes equations. Recalling the momentum equation of (2.26) it follows

$$\begin{aligned} \rho\big(\partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\big) - \mu_{\mathrm{eff}}\Delta\boldsymbol{u} + \nabla p &= -\mu K^{-1}\boldsymbol{u} \\ &\stackrel{\mathrm{def}}{=} \boldsymbol{f}_{\mathrm{por}}. \end{aligned} \tag{2.83}$$

Thus the presence of porous media can be seen as a forcing term in the Stokes equations with $\boldsymbol{f}_{\mathrm{por}} = -\mu K^{-1}\boldsymbol{u}$.

In lattice Boltzmann methods, there exists a number of schemes to consider body forces. For an overview and analysis of different schemes, see for example [59] and [93]. In the context of porous media flow, the work of Spaid and Phelan [70], and Guo and Zhao [71] will be considered here.

Spaid and Phelan model

Spaid and Phelan [70] were the first to propose a method to model porous media flow in lattice Boltzmann methods. They used a forcing scheme proposed by Shan and Shen [94], which was initially considered to simulate multiphase flow, but can also be used as general forcing scheme.

The basic idea of this scheme is that a force $\boldsymbol{f} : I \times \Omega \to \mathbb{R}^d$, changes the velocity of the fluid by Newton's second law of motion, thus

$$\begin{aligned} \boldsymbol{u}^{\mathrm{eq}}(t, \boldsymbol{x}) &= \frac{1}{\rho(t, \boldsymbol{x})} \sum_{i=0}^{q-1} \boldsymbol{c}_i f_i(t, \boldsymbol{x}) + \frac{\tau}{\rho(t, \boldsymbol{x})} \boldsymbol{f}(t, \boldsymbol{x}) \\ &= \boldsymbol{u}(t, \boldsymbol{x}) + \triangle \boldsymbol{u}(t, \boldsymbol{x}), \end{aligned} \tag{2.84}$$

where $\boldsymbol{u}^{\mathrm{eq}}$ enters the equilibrium distribution function by

$$f_i^{\mathrm{eq}}(\rho, \boldsymbol{u}^{\mathrm{eq}}) = w_i \rho \left(1 + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}^{\mathrm{eq}}}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}^{\mathrm{eq}})^2}{2c_s^4} - \frac{\boldsymbol{u}^{\mathrm{eq}} \cdot \boldsymbol{u}^{\mathrm{eq}}}{2c_s^2} \right). \tag{2.85}$$

Here only the dependency on the moments is shown. Using the force of the porous medium $\boldsymbol{f} = -\mu K^{-1} \boldsymbol{u}$, it follows

$$\begin{aligned} \boldsymbol{u}^{\mathrm{eq}} &= \boldsymbol{u} - \frac{\tau}{\rho} \mu K^{-1} \boldsymbol{u} \\ &= (1 - \tau \nu K^{-1}) \boldsymbol{u} \\ &= d \boldsymbol{u}, \end{aligned} \tag{2.86}$$

with $\nu \overset{\mathrm{def}}{=} \frac{\mu}{\rho}$ the kinematic viscosity. Here $d(\cdot, \boldsymbol{x}) \overset{\mathrm{def}}{=} 1 - \tau \nu K^{-1}(\cdot, \boldsymbol{x})$ has a value of $d \in [0, 1]$. For $d(\cdot, \boldsymbol{x}) = 0$ this results in a pure solid node. And for $d(\cdot, \boldsymbol{x}) = 1$, i.e. standard equilibrium distribution function, this results in pure fluid node. Because of these properties d is called *lattice-porosity*, although it should not be seen as physical porosity [2]. Additionally, an extension of the method was presented by Martys [48] to account for the usage of different effective viscosities.

Based on the fact that this forcing only needs to change the velocity of the equilibrium distribution function makes this scheme fast and easy to implement, and has thus been used also for particle simulations [76] or topology optimisation [12].

Guo and Zhao model

Another method was proposed by Guo and Zhao [71], who used a different forcing scheme, also proposed by Guo et al. [95], where the equilibrium distribution function gets updated by

$$\boldsymbol{u}^{\mathrm{eq}}(t,\boldsymbol{x}) = \frac{1}{\rho(t,\boldsymbol{x})}\sum_{i=0}^{q-1}\boldsymbol{c}_i f_i(t,\boldsymbol{x}) + \frac{\triangle t}{2\rho(t,\boldsymbol{x})}\boldsymbol{f}(t,\boldsymbol{x}). \tag{2.87}$$

In addition to the force entering the equilibrium distribution function, the collision term is also adjusted, reading

$$f_i(t+\triangle t, \boldsymbol{x}+\boldsymbol{c}_i\triangle t) = f_i(t,\boldsymbol{x}) - \frac{\triangle t}{\tau}\Big(f_i(t,\boldsymbol{x}) - f_i^{\mathrm{eq}}(t,\boldsymbol{x})\Big) + F_i(t,\boldsymbol{x}), \tag{2.88}$$

where F_i is proposed as [95]

$$F_i(t,\boldsymbol{x}) = \triangle t\left(1-\frac{\triangle t}{2\tau}\right) w_i\left(\frac{\boldsymbol{c}_i - \boldsymbol{u}(t,\boldsymbol{x})}{c_s^2} + \frac{\big(\boldsymbol{c}_i\cdot\boldsymbol{u}(t,\boldsymbol{x})\big)\boldsymbol{c}_i}{c_s^4}\right)\cdot\boldsymbol{f}(t,\boldsymbol{x}). \tag{2.89}$$

For the application to porous media another difference to the model of Spaid and Phelan is that now the equilibrium velocity is implicitly given. Thus, the force is described by $\boldsymbol{f} = -\mu K^{-1}\boldsymbol{u}^{\mathrm{eq}}$, and it follows

$$\begin{aligned}\boldsymbol{u}^{\mathrm{eq}} &= \frac{1}{\rho}\sum_{i=0}^{q-1}\boldsymbol{c}_i f_i - \frac{\triangle t}{2\rho}\mu K^{-1}\boldsymbol{u}^{\mathrm{eq}} \\ &= \frac{\boldsymbol{u}}{1+\frac{\triangle t}{2}\nu K^{-1}},\end{aligned} \tag{2.90}$$

where $\boldsymbol{u} = \frac{1}{\rho}\sum_{i=0}^{q-1}\boldsymbol{c}_i f_i$ is a temporal velocity [71].

This model is more general than the one of Spaid and Phelan, as it can account for different porosities, and include the Forchheimer term [71].

2.4 Numerical experiments of porous media flow

Here the porous media models above will be tested for accuracy and performance, as these models will be the basis for the topology optimisation used in this thesis. For this purpose, a test case of the flow of porous media in a pipe will be constructed, for which an analytical solution is derived here.

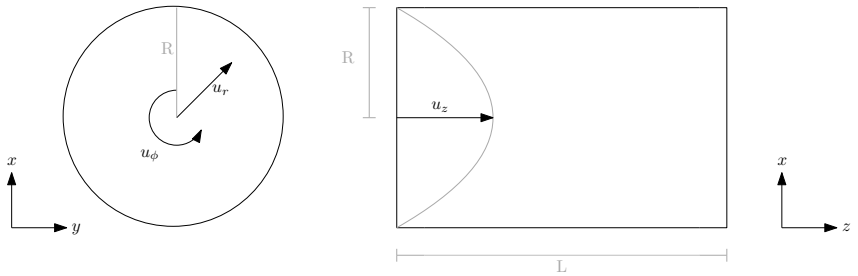


Figure 2.6: Pipe flow with velocity in cylindrical coordinates, i.e. $\boldsymbol{u} = (u_r, u_\phi, u_z)^\top$.

2.4.1 Test case and analytical solution

Here a steady state, low Reynolds number flow through porous media with no-slip boundary condition is considered, and thus

$$-\mu\Delta\boldsymbol{u} + \nabla p + \mu K^{-1}\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{2.91a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{2.91b}$$

$$\boldsymbol{u} = 0 \qquad \text{on } \partial\Omega. \tag{2.91c}$$

Without the porous medium, i.e. $\mu K^{-1}\boldsymbol{u} \equiv 0$, this case reduces to a Poiseuille flow.

In order to obtain an analytical solution in three dimensions (2.91) is transferred to cylindrical coordinates, i.e. for $\boldsymbol{x} = \big(x, y, z\big)^\top$ it follows

$$x = r\sin(\phi), \qquad y = r\cos(\phi), \qquad z = z, \tag{2.92}$$

where $r = \sqrt{x^2 + y^2 + z^2}$ is the radius, and ϕ the angle. Let $\boldsymbol{u} = (u_r, u_\phi, u_z)^\top$ be the velocity in cylindrical coordinates (cf. Figure 2.6). For a flow in z-direction it follows $u_r = u_\phi = 0$ and thus

$$\nabla \cdot \boldsymbol{u} = 0 \;\Rightarrow\; \partial_z u_z = 0. \tag{2.93}$$

With this it follows $\partial_r p = \partial_\phi p = 0$, and the momentum equation reduces to

$$-\mu\left(\frac{\mathrm{d}^2}{\mathrm{d}r^2}u_z(r) + \frac{1}{r}\frac{\mathrm{d}}{\mathrm{d}r}u_z(r)\right) + \mu K^{-1}u_z(r) + \frac{\mathrm{d}p}{\mathrm{d}z} = 0, \tag{2.94}$$

where the velocity is therefore only dependent on the radius and has only the z-component.

An ansatz to this differential equation is

$$u_z(r) = C_1 I_0\left(\frac{r}{\sqrt{K}}\right) + C_2 Y_0\left(\frac{ir}{\sqrt{K}}\right) - \frac{K}{\mu}\frac{\mathrm{d}p}{\mathrm{d}z}, \tag{2.95}$$

with $C_1, C_2 \in \mathbb{R}$ constants of integration, and I_0 and Y_0 the Bessel functions of first and second kind, respectively. For $r = 0$ it follows $I_0(r) = 1$ and $\lim_{r\to 0} Y_0(r) = -\infty$, and hence $C_2 = 0$. And with $u(R) = 0$, i.e. no-slip at the boundary, it follows for the analytical solution

$$u_z(r) = \frac{K}{\mu}\frac{\mathrm{d}p}{\mathrm{d}z}\left(1 - \frac{I_0\left(rK^{-\frac{1}{2}}\right)}{I_0\left(RK^{-\frac{1}{2}}\right)}\right), \tag{2.96}$$

with modified Bessel function of first kind I_0

$$I_0(z) = \sum_{k=0}^{\infty} \frac{\left(\frac{1}{4}z^2\right)^k}{(k!)^2}. \tag{2.97}$$

Note, for the analytical solution in two dimensions the modified Bessel function is replaced by the hyperbolic cosine function [71, 96].

The test case considered has following properties. Viscosity of $\mu \in \{1.0, 0.1\}$ Pa s, density of $\rho = 1.0\ \frac{\mathrm{kg}}{\mathrm{m}^3}$, length of $L = 2.0$ m, radius of $R = 0.5$ m (cf. Figure 2.6) and inflow velocity of $U = 1.0\ \frac{\mathrm{m}}{\mathrm{s}}$, which results in a Reynolds number of $Re \in \{1.0, 10.0\}$. For LBM a relaxation time of $\tau = 0.8$ is chosen. The problem is considered in three dimensions, and for the numerical experiment the modified Bessel function of first kind is approximated here by

$$I_0(z) \approx \frac{\cosh(z)}{(1 + \frac{1}{4}z^2)^{\frac{1}{4}}} \frac{1 + 0.24273z^2}{1 + 0.43023z^2}, \tag{2.98}$$

which was shown to have a maximum relative error of less than 6.0×10^{-3} (cf. [97]).

2.4.2 Results

The results of the test case are analysed using a convergence test with L^1, L^2 and L^∞ norm, with

$$||\boldsymbol{u}||_{L^1(\Omega)} = \int_\Omega |\boldsymbol{u}|\,\mathrm{d}\boldsymbol{x}, \quad ||\boldsymbol{u}||_{L^2(\Omega)} = \left(\int_\Omega |\boldsymbol{u}|^2\,\mathrm{d}\boldsymbol{x}\right)^{\frac{1}{2}}, \quad ||\boldsymbol{u}||_{L^\infty(\Omega)} = \max_\Omega |\boldsymbol{u}|.$$

The error measured is the relative difference of the analytical $\boldsymbol{u}^*$ to the numerical velocity field $\boldsymbol{u}_f$ for two different permeability values and Reynolds numbers, shown in Figure 2.7. The results show similar results for the Spaid and Phelan, as well as, the Guo and Zhao

model, with second order convergence for the error measured in L^1 and L^2 norm, as expected for the interpolated bounce back boundary condition used here.

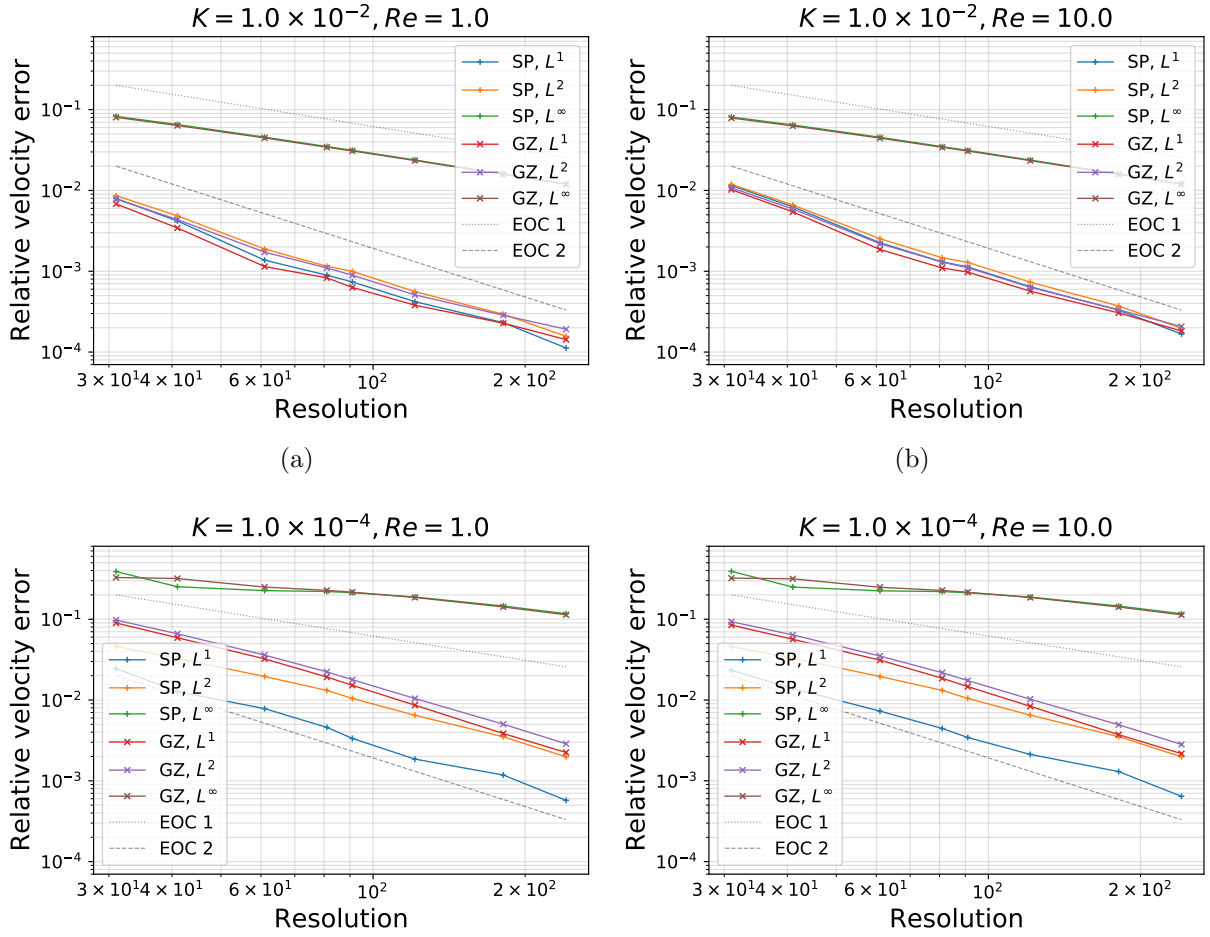
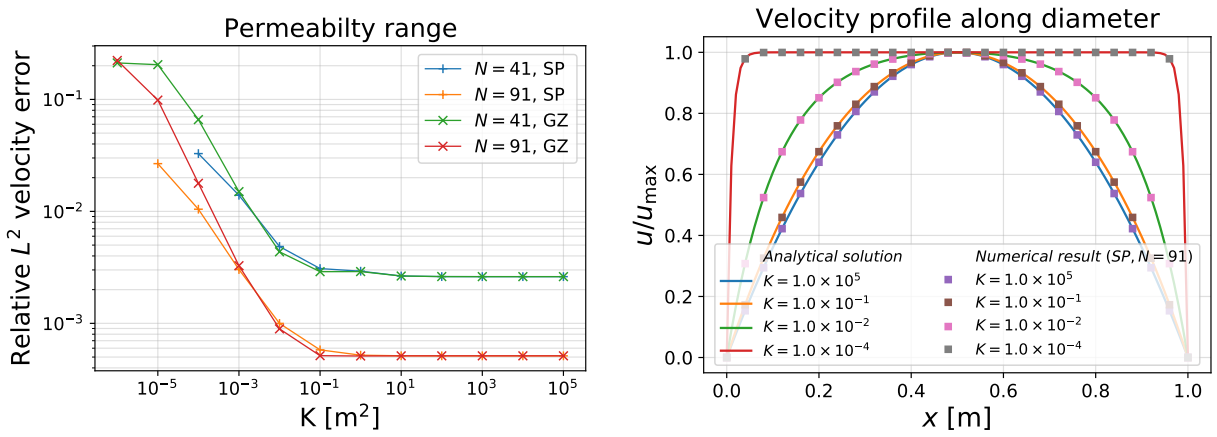


Figure 2.7: Convergence test for the porous media test case for different Reynolds numbers and permeabilities.

In Figure 2.8 the results for different permeability values are presented. As expected, the error is lowest for Poiseuille flow, i.e. $K \to \infty$, and grows larger the more solid the porous medium becomes, i.e. $K \to 0$. Although the model of Guo and Zhao can simulate lower permeability values than the model of Spaid and Phelan these have high errors. The comparison of numerical and analytical flow profiles shows a very high agreement for different permeability values, see Figure 2.8b.

The results in this case are nearly identical for the Spaid and Phelan and the Guo and Zhao model, which was also shown by Huang et al. [93] albeit for a different application using the underlying forcing schemes of Shan and Chen [94] and Guo et al. [95]. The performance however, is much higher for the Spaid and Phelan model with only half the simulation time. Thus, for this thesis the Spaid and Phelan model is very well suited.



(a) Permeability range

(b) Velocity profiles

Figure 2.8: Error analysis for porous media test case with respect to the permeability of the medium for a Reynolds number of $Re = 1$. Comparison between Spaid and Phelan and Guo and Zhao model for resolutions of $N = 41$ and $N = 91$ for different permeabilities on the left. And comparison of analytical to numerical velocity profiles for the Spaid and Phelan model with $N = 91$ on the right.

Chapter 3

Magnetic resonance imaging

Magnetic resonance imaging is a highly valuable diagnostic tool, especially known in medical imaging to visualise the organs, tissues, bones, etc. of patients. MRI is based on the effect of nuclear magnetic resonance (NMR), where nuclei with non-zero spin, like e.g. ^{1}H, ^{19}F, or ^{31}P have a magnetic moment. If these nuclei are placed inside a strong magnetic field they can absorb and emit radio frequency which can be measured. Since objects with different properties emit to different degrees, an image of the interior can be created. For a deep insight into the functioning and underlying physical laws, the books of Callaghan [98] and Reiser et al. [99] are recommended.

In this chapter, the basics of MRI for the measurement of fluid flows, as well as the problem of measurement noise and its mathematical description are briefly introduced. The latter in particular is important for the generation of synthetic data to validate the method with regard to noise reduction in Chapter 8.

3.1 Flow MRI

MRI is able to image velocity fields, which is called magnetic resonance velocimetry [100], or *flow MRI* [101]. With this it is possible to diagnose cardiac diseases by measuring the blood flow [101], and it is even capable to image turbulent and multiphase flows [100, 102]. Another application is the analysis of flow in porous media [100, 103, 104], which is often used in oil recovery industry or water purification [102]. For an overview of flow MRI applications the works of Elkins and Alley [100], and Gladden [102] are recommended.

One technique of flow imaging is thereby the phase contrast MRI, which can be used with conventional MRI scanners without the use of tracers, particles or the like [100, 102, 105].

The basic idea of phase contrast MRI is the usage of bipolar gradients, which have a different effect on spins of moving nuclei and was proposed by Moran in 1982 [105].

The phase shift Φ of the nuclei can be calculated by [100, 106–108]

$$\Phi = \gamma \int_{t}^{t+\triangle t} \boldsymbol{G}(t') \cdot \boldsymbol{x}(t') \, \mathrm{d}t', \tag{3.1}$$

where $\boldsymbol{G}$ is the magnetic field gradient, and γ the gyromagnetic ratio of the observed nucleus. Expanding the time dependent position vector it follows

$$\boldsymbol{x}(t') = \boldsymbol{x}_0 + \boldsymbol{v}_0 t' + \mathcal{O}(t'^2), \tag{3.2}$$

where $\boldsymbol{x}_0$ and $\boldsymbol{v}_0$ are the position and velocity of the nucleus, respectively. Defining the gradient moments as $\boldsymbol{M}_i \overset{\text{def}}{=} \int t^i \boldsymbol{G}(t) \, \mathrm{d}t$, and omitting the second order terms in the expansion, it follows

$$\Phi \approx \gamma \boldsymbol{M}_0 + \boldsymbol{v}_0 \cdot \gamma \boldsymbol{M}_1. \tag{3.3}$$

Using two inverse, bipolar gradients, such that $\boldsymbol{M}_0 = 0$ (cf. [100, 108]), it follows that the phase difference $\triangle\Phi$ can be calculated as

$$\triangle\Phi \approx \boldsymbol{v}_0 \cdot \gamma \triangle \boldsymbol{M}_1, \tag{3.4}$$

where $\triangle \boldsymbol{M}_1 = 2\boldsymbol{M}_1$ due to the bipolar field [108]. The measurable phase difference is therefore proportional to the velocity.

In measurements it is common to use a parameter, called velocity encoding v_{enc}, which is the velocity that results in a phase shift of π radians (or 180°) [109]. This parameter should be set to the highest expected velocity, as it will otherwise lead to aliasing effects [107, 109], but as low as possible to reduce measurement noise (cf. Section 3.2). Using a gradient in one direction and let v and M_1 be the corresponding components in this direction it follows that the velocity can be calculated by

$$v \approx \frac{v_{\text{enc}}}{\pi} \triangle\Phi, \tag{3.5}$$

where

$$v_{\text{enc}} \overset{\text{def}}{=} \frac{\pi}{\gamma \triangle M_1}. \tag{3.6}$$

3.2 Measurement noise

A major problem with MRI measurements is the presence of statistical measurement noise [110]. There are many reasons for this noise, including inhomogeneities of the magnetic fields, movements of the observed objects or thermal noise of the electrical components [110, 111]. However, it is possible to describe the noise by certain probability functions [110–115].

In general, the noise of MR images has been shown to be governed by a Rician distribution [112]

$$p_{\text{Rice}}(x) = \frac{x}{\sigma^2} \exp\left(-\frac{(x^2+S^2)}{2\sigma^2}\right) I_0\left(\frac{xS}{\sigma^2}\right), \tag{3.7}$$

where S is the (noise free) signal, σ the noise, and I_0 the Bessel function of the first kind. In the absence of a signal, i.e. $S = 0$, the Rician distribution simplifies to the Rayleigh distribution

$$p_{\text{Rayleigh}}(x) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{3.8}$$

A common parameter in MRI that quantifies the image quality is the signal-to-noise ratio (SNR) defined as

$$\text{SNR} \stackrel{\text{def}}{=} \frac{S}{\sigma}. \tag{3.9}$$

The higher the SNR the better the quality of the image. Also, it can be shown that the Rician distribution (3.7) tends to be a Gaussian, or normal distribution for higher SNR [110–113]

$$p_{\text{Gauss}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\sqrt{S^2+\sigma^2})^2}{2\sigma^2}\right), \tag{3.10}$$

with mean of $\sqrt{S^2+\sigma^2}$ and variance of σ. This tendency is shown in Figure 3.1, where a SNR of two approximates the Gaussian distribution very well. This relation will be used in Chapter 8.

It can be shown that the SNR of an MRI measurement is determined by at least two factors [98, 113]. First, by the *spatial resolution*

$$\frac{S}{\sigma} \propto V_S, \tag{3.11}$$

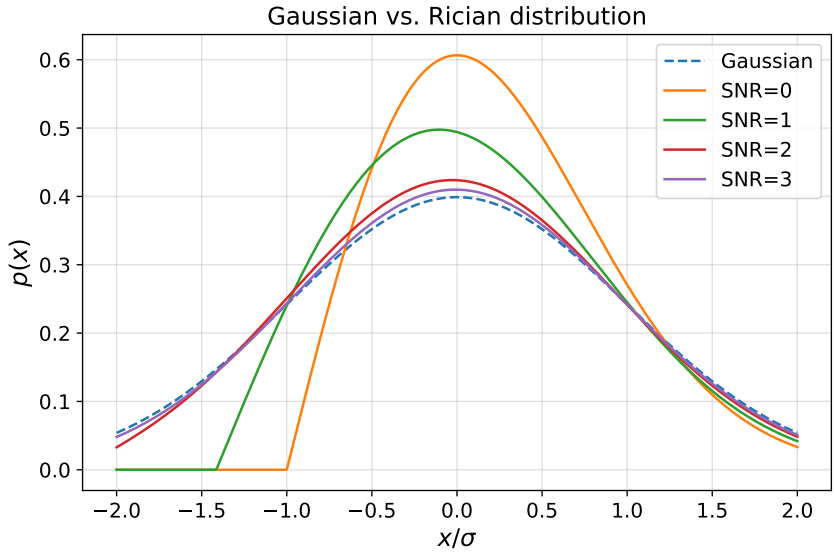


Figure 3.1: Comparison of Rician (3.7) to Gaussian distribution (3.10) for different signal-to-noise ratios (SNR) with mean at zero. The higher the SNR the more the Rician distribution follows a Gaussian one.

which depends on the voxel size V_S, which is the three-dimensional equivalent to a pixel, i.e. the smallest discrete element of the image. And second, by the measurement time, i.e. *number of acquisitions*

$$\frac{S}{\sigma} \propto \sqrt{N_{\text{acc}}}. \tag{3.12}$$

If the resolution is increased, i.e. voxel size reduced, the MRI is able to identify smaller structures, but with the consequence of a decreased SNR and thus an increase in noise. If, on the other hand, the number of acquisitions and hence the SNR is increased, the measurement time is longer, which is often a limiting factor for the observation of certain processes, e.g. filtration and fouling [116]. Further dependencies on the SNR are the magnetic field strength, which should be chosen as strong as possible. Or the number of spins, which explains the common use of ^{1}H for imaging, because of its high abundance in nature, but also due to its intrinsically high SNR thanks to its gyromagnetic factor [100].

The signal-to-noise ratio in flow MRI, SNR_v, is related to the SNR of the stationary MRI by [107, 108, 117]

$$\text{SNR}_v = \frac{\pi}{\sqrt{2}} \frac{v}{v_{\text{enc}}} \text{SNR}. \tag{3.13}$$

Due to this relation the velocity encoding parameter in flow MRI should be set as low as possible, but not too low because of velocity aliasing effects (cf. Section 3.1).

Chapter 4

Optimal control

Optimisation is used in almost all parts of science and industry. It is used to reduce weight while at the same time increasing stability of parts in aircrafts or cars, to reduce time and distances of deliveries in logistics, or to increase the accuracy of weather forecasts. And it is used in artificial intelligence and machine learning.

In the following, a general definition of optimisation problems, its solutions and its characterisation, i.e. necessary conditions, are given. This information can then be used for solving strategies, like gradient based methods, for which different conditions are needed, namely the choice of step sizes, and search directions. The section will conclude with an algorithm, which is used in the numerical validation and applications considered in this thesis. A major part is the calculation of gradients, for which different techniques are presented, the most important for this work being the adjoint approach.

This chapter is mainly based on the work of Tröltzsch [118], Hinze et al. [27], De los Reyes [119], and Nocedal and Wright [120].

4.1 General optimisation problems

Here, the general definition of an optimisation problem, optimal solutions and their existence are presented. In order to keep these definitions general, they are defined in function spaces, but if suitable and helpful for understanding, they are also transfereed to finite dimensional problems.

Let $J : Y \times U \to \mathbb{R}$, $\boldsymbol{R} : Y \times U \to Z$ be operators between Banach spaces Y, U, Z. A

general constrained optimisation problem is given by

$$\min_{(f,\alpha)\in Y_{ad}\times U_{ad}} J(f,\alpha) \quad \text{subject to} \quad \boldsymbol{R}(f,\alpha) = 0, \tag{4.1}$$

where J is called goal or *objective function*, $\boldsymbol{R}(f,\alpha) = 0$ *side condition* or state equation, $f \in Y$ the *state* and $\alpha \in U$ the *control* or design variable, with $U_{ad} \subset U$ and $Y_{ad} \subset Y$ called *admissible sets*. If the side condition is governed by a partial differential equation, this problem is called *optimal control problem*.

Assuming there exists a solution operator for the side condition $\boldsymbol{R}(f,\alpha) = 0$ which gives a unique solution $f(\alpha) \in Y$ for every $\alpha \in U$, and defining $J(f,\alpha) = J\left(f(\alpha),\alpha\right) \overset{\text{def}}{=} J(\alpha)$, the *reduced problem*

$$\min_{\alpha\in U_{ad}} J(\alpha) \tag{4.2}$$

can be considered.

Now the question arises, what is an optimal solution and how can it be characterised? This is addressed in the following.

4.1.1 Optimal solutions and optimality conditions

Next, the definition of an *optimal solution* of the reduced problem (4.2), i.e. a minimum, is given.

Definition 4.1 (Optimal solution)**.** *For a (global) optimal solution $\bar{\alpha} \in U_{ad}$ it holds*

$$J(\bar{\alpha}) \leq J(\alpha) \quad \forall \alpha \in U_{ad}. \tag{4.3}$$

For a local optimal solution $\bar{\alpha}$ the above is true for some neighbourhood $\mathcal{N}(\bar{a}) \subset U_{ad}$ of $\bar{\alpha}$.

In order to characterise such solutions, a necessary condition for an optimal solution is presented. This will be especially helpful for finding optimal solutions numerically (cf. Section 4.2).

For this, definitions for derivatives in function spaces must first be introduced. Let $F : U \subset X \to Y$ be an operator between Banach spaces X, Y and $U \neq \emptyset$ open.

Definition 4.2 (Directional derivative)**.** *F is called* directionally differentiable *at $u \in U$*

if the limit

$$\delta F(u,h) \stackrel{\text{def}}{=} \lim_{t\to 0} \frac{1}{t}\big(F(u+th) - F(u)\big) \tag{4.4}$$

exists for all $h \in X$*. Then* $\delta F(u,h)$ *is called* directional derivative *of* F *in direction* h*.*

Definition 4.3 (Gâteaux derivative)**.** *F is called Gâteaux differentiable at $u \in U$ if F is directionally differentiable at u and there exists $A \in \mathcal{L}(U,V)$ with $h \mapsto Ah \stackrel{\text{def}}{=} \delta F(u,h)$. Then $F'(u) \stackrel{\text{def}}{=} A$ is called Gâteaux derivative of F at u.*

Definition 4.4 (Fréchet derivative)**.** *F is called Fréchet differentiable at $u \in U$ if F is Gâteaux differentiable at u and following holds*

$$F(u+h) = F(u) + F'(u)h + r(u,h), \tag{4.5}$$

with $\frac{||r(u,h)||_V}{||h||_U} \to 0$*, for* $||h||_U \to 0$*.*
Then $F'(u)$ is called Fréchet derivative *of F at u.*

Note, every Fréchet differentiable operator is also Gâteaux differentiable and the derivatives are identical [118]. Also, every linear, bounded operator $A \in \mathcal{L}(X)$ is Fréchet differentiable and the derivative is the operator itself, as $A(x+h) = A(x) + A(h) + \underbrace{A(0)}_{=0}$.

For Fréchet derivatives the chain rule holds [118].

Definition 4.5 (Chain rule)**.** *Let $X \subset U, Y \subset V$ be open sets and $F : X \to Y$ and $G : X \to Z$ Fréchet differentiable at $x \in X$ and $F(x)$ for Banach spaces U, V, Z. Then $H = G \circ F$ is Fréchet differentiable at x and*

$$H'(x) = G'(F(x))F'(x). \tag{4.6}$$

If F in the above definition is only Gâteaux differentiable it follows that H is Gâteaux differentiable. Therefore, the sum rule holds for Fréchet and Gâteaux differentials [27].

With the above definitions, a necessary condition for an optimal solution can be derived.

Lemma 4.1 (Optimality condition)**.** *Let U be a real Banach space, $U_{ad} \subset U$ a convex set and J a Gâteaux differentiable real valued functional on an open subset of U_{ad}. Suppose $\bar{\alpha} \in U_{ad}$ is a solution of*

$$\min_{\alpha \in U_{ad}} J(\alpha),$$

then the following variational inequality, called optimality condition*, holds*

$$J'(\bar{\alpha})(\alpha - \bar{\alpha}) \geq 0 \quad \forall \alpha \in U_{ad}. \tag{4.7}$$

Proof. For an arbitrary $t \in (0, 1]$ define $\alpha(t) = \bar{\alpha} + t(\alpha - \bar{\alpha})$, with $\alpha(t) \in U_{ad}$ because of convexity of U_{ad}. Further, as $\bar{\alpha}$ is the optimal solution, it holds $J(\bar{\alpha}) \leq J(\alpha)$ for all $u \in U_{ad}$. Therefore, it follows

$$\frac{1}{t}\big(J(\bar{\alpha} + t(\alpha - \bar{\alpha})) - J(\bar{\alpha})\big) \geq 0.$$

Taking the limit for t to zero completes the proof. □

If J is a convex function, the optimality condition (4.7) is also sufficient, see for example Tröltzsch [118], Lemma 2.21. Note that for $U_{ad} = U$, and especially $U_{ad} = \mathbb{R}^n$, the well known optimality condition

$$J'(\bar{\alpha}) = 0 \tag{4.8}$$

holds true [119].

4.1.2 Existence of optimal solutions

The requirements which have to be imposed on the optimisation problem in order to make statements about the existence of optimal solutions, can be very different in their complexity. This depends not only on the objective function and the constraints, but also on whether the problem is considered in finite or infinite dimensional spaces.

First, a finite dimensional problem is considered.

Theorem 4.1 (Existence of optimal solutions in $\mathbb{R}^n$)**.** *Let $U_{ad} \subset \mathbb{R}^n$ nonempty, closed and bounded, $J : \mathbb{R}^n \to \mathbb{R}$ continuous, then the problem $\min\limits_{\alpha \in U_{ad}} J(\alpha)$ has an optimal solution.*

Proof. The set U_{ad} is closed and bounded and therefore compact, following the Weierstraß extreme value theorem the continuous function J attains a minimum (and a maximum) on U_{ad}. □

For optimisation problems in function spaces the existence of optimal solutions is much more complex, due to the fact that from closed and boundedness does not follow compactness of the set and therefore the extreme value theorem can not be applied. Thus,

the following definition of weak convergence is needed, see for example Tröltzsch [118] and Hinze et al. [27].

Definition 4.6 (Weak convergence)**.** *Let X be a real Banach space and X^* its dual space. A sequence $(x_k)_{k=1}^{\infty} \in X$ is called* weakly convergent *to $x \in X$, if for $k \to \infty$ it holds*

$$F(x_k) \to F(x) \quad \forall F \in X^*.$$

The weak convergence *is written as $x_k \rightharpoonup x$.*

With this, statements can now be made about the existence of optimal solutions for non-linear optimisation problems.

Theorem 4.2 (Existence of optimal solutions)**.** *Let $J : Y \times U \to \mathbb{R}$, $\boldsymbol{R} : Y \times U \to Z$ be continuous with Banach space Z, reflexive Banach spaces U, Y and*

a) $U_{ad} \subset U$ is convex, closed and bounded

b) $Y_{ad} \subset Y$ is convex and closed

c) $U_{ad} \ni \alpha \mapsto f(\alpha) \in Y$ is a bounded solution operator

d) $Y \times U \ni (f, \alpha) \mapsto \boldsymbol{R}(f, \alpha) \in Z$ is continuous under weak convergence

e) For $\alpha_k \rightharpoonup \alpha$ it follows $\liminf_{k\to\infty} J(\alpha_k) \geq J(\alpha)$, i.e. J is weakly semicontinuous.

Then there exists an optimal solution for (4.1)*.*

Proof. The proof can be found in Hinze et al. [27], Theorem 1.45. □

Note, if the side condition $\boldsymbol{R}(f, \alpha) = 0$ is governed by a linear problem, the assumptions of Theorem 4.2 can be significantly reduced (cf. [27]). The main difficulty comes from the non-linearity of the side condition in PDE-constrained optimisation problems.

Next, solution strategies for optimal control problems are considered.

4.2 Gradient based optimisation

Many methods have been developed to solve optimisation problems. Most efficient methods are gradient based, like trust-region, sequential quadratic problem (SQP), or line search [120]. Here, the focus lies in the gradient based line search methods.

For the sake of simplicity, only finite dimensional problems are considered in this section. Let $Y \subset \mathbb{R}^m$, $U \subset \mathbb{R}^n$ and $Z \subset R^l$, and J sufficiently smooth for the optimisation problem (4.2). Further, let $\nabla J(\boldsymbol{\alpha}) = J'(\boldsymbol{\alpha})^\top$ be the column vector gradient.

4.2.1 Line search method

The line search method is a family of iterative algorithms to find local minima. The basic idea is to start from an initial guess $\boldsymbol{\alpha}_0$ and find a sequence $(\boldsymbol{\alpha}_k)_{k\in\mathbb{N}_0}$, for which holds

$$J(\boldsymbol{\alpha}_{k+1}) < J(\boldsymbol{\alpha}_k), \tag{4.9}$$

where the next control $\boldsymbol{\alpha}_{k+1}$ is found searching along a line

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \lambda_k \boldsymbol{p}_k. \tag{4.10}$$

Here $\lambda_k \in \mathbb{R}$ is called *step size* and $\boldsymbol{p}_k \in \mathbb{R}^n$ the *search direction*.

There exists an optimal step size

$$\lambda_{opt} = \underset{\lambda>0}{\operatorname{argmin}}\ \Phi(\lambda) \overset{def}{=} J(\boldsymbol{\alpha}_k + \lambda \boldsymbol{p}_k),$$

but since the calculation of the optimal step size can be as complex as finding the optimal solution, the *inexact line search* method has developed. It uses a predefined initial step size and checks certain conditions that the step size must fulfil. If this is not the case, the step size is increased or decreased depending on the step condition.

The general algorithm of an inexact line search is given as

Algorithm 1 Inexact line search

Set $k = 0$
Choose initial guess $\boldsymbol{\alpha}_0$ and tol > 0
repeat
  Compute *search direction* $\boldsymbol{p}_k$
  Choose λ_k fulfilling a *step condition*
  Set $\boldsymbol{\alpha}_{k+1} \leftarrow \boldsymbol{\alpha}_k + \lambda_k \boldsymbol{p}_k$
  Set $k \leftarrow k + 1$
until $||\nabla J(\boldsymbol{\alpha}_k)|| <$ tol

The condition $||\nabla J(\boldsymbol{\alpha}_k)|| <$ tol of Algorithm 1 is the numerical version of the necessary condition (4.8) of the optimisation problem.

Next, the choices of the step condition and the search direction are investigated.

4.2.2 Step condition

The simplest step condition is checking if the objective function is decreasing, i.e. $J(\boldsymbol{\alpha}_{k+1}) < J(\boldsymbol{\alpha})$, but this condition can be to weak, as a small step size can result in a very slow convergence [120]. Therefore, more sophisticated step conditions have evolved, one of them being the famous *Wolfe conditions* [121, 122].

The Wolfe conditions consist of two conditions, namely the *Armijo condition*

$$J(\boldsymbol{\alpha}_k + \lambda_k \boldsymbol{p}_k) \leq J(\boldsymbol{\alpha}_k) + c_1 \lambda_k \nabla J(\boldsymbol{\alpha}_k)^\top \boldsymbol{p}_k, \tag{4.11}$$

which checks sufficient decrease of the objective function, and the *curvature condition*

$$\nabla J(\boldsymbol{\alpha}_k + \lambda_k \boldsymbol{p}_k)^\top \boldsymbol{p}_k \geq c_2 \nabla J(\boldsymbol{\alpha}_k)^\top \boldsymbol{p}_k, \tag{4.12}$$

which ensure that the step length is large enough. The curvature condition has more positive properties, which will be presented in Section 4.2.3. For the constants c_1 and c_2, which help tuning the Wolfe conditions, it must hold $0 < c_1 < c_2 < 1$. Typical values are $c_1 = 10^{-4}$ and $c_2 = 0.9$ [120].

Figure 4.1 illustrates the Wolfe conditions for an arbitrary function J. Each point of the plot shown would fulfil the condition that the function value should be reduced. The Wolfe conditions, on the other hand, reduce the range of admissible step sizes to a smaller and more meaningful range.

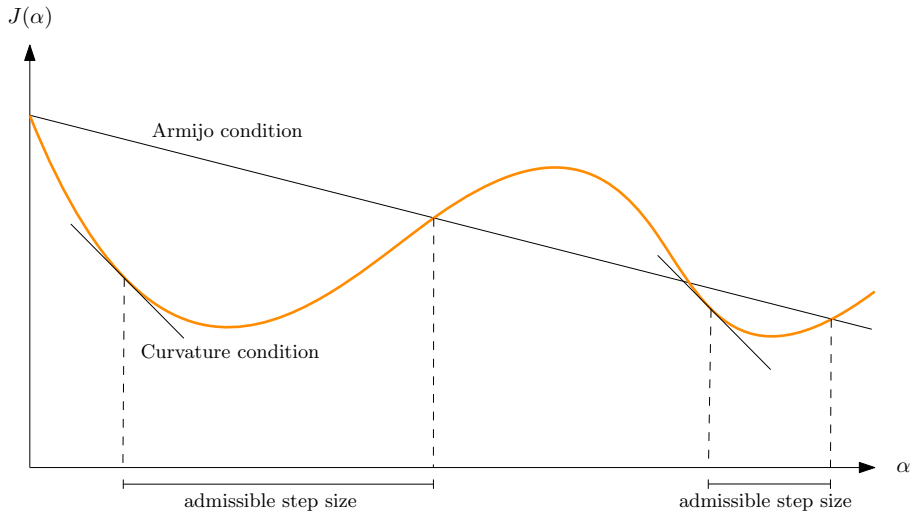


Figure 4.1: Illustration of the Wolfe conditions, consisting of the Armijo condition (sufficient decrease) and curvature condition (large enough step). If only a decrease of the function would be required, every point on the shown axis would be admissible. This illustration is based on the original figure of Nocedal and Wright [120].

4.2.3 Search direction

The search direction $\boldsymbol{p}_k$ is usually selected so that the function is reduced along this direction. Therefore, most line search methods require the search direction to be a descent direction.

Definition 4.7 (Descent direction)**.** *Let $J : \mathbb{R}^n \to \mathbb{R}$ and $\boldsymbol{\alpha} \in \mathbb{R}^n$. If there exists a $\lambda_0 > 0$ with $J(\boldsymbol{\alpha} + \lambda \boldsymbol{p}) < J(\boldsymbol{\alpha})$ for all $\lambda \in (0, \lambda_0]$, then $\boldsymbol{p} \in \mathbb{R}^n$ is called* descent direction*.*

The easiest and simplest descent direction is the choice of the gradient $-\nabla J(\boldsymbol{\alpha}_k)$, which can be seen after using the Taylor expansion $J(\boldsymbol{\alpha}_k + \lambda_k \boldsymbol{p}_k) \approx J(\boldsymbol{\alpha}_k) + \lambda_k \nabla J(\boldsymbol{\alpha}_k)^\top \boldsymbol{p}_k$ and

$$\nabla J(\boldsymbol{\alpha}_k)^\top \boldsymbol{p}_k = -\nabla J(\boldsymbol{\alpha}_k)^\top \nabla J(\boldsymbol{\alpha}_k) = -||\nabla J||^2 \leq 0.$$

Thus, a descent direction can be characterised by ensuring $\nabla J(\boldsymbol{\alpha}_k)^\top \boldsymbol{p}_k < 0$. Generally, a search direction has the following form

$$\boldsymbol{p}_k = -\mathbf{B}_k^{-1} \nabla J(\boldsymbol{\alpha}_k), \tag{4.13}$$

with $\mathbf{B}_k$ symmetric and invertible. Then a descent direction

$$\boldsymbol{p}_k^\top \nabla J(\boldsymbol{\alpha}_k) = -\nabla J(\boldsymbol{\alpha}_k)^\top \mathbf{B}_k^{-1} \nabla J(\boldsymbol{\alpha}_k) \leq 0, \tag{4.14}$$

can be ensured for $\mathbf{B}_k^{-1}$ positive definite.

Depending on $\mathbf{B}_k$ different methods can be defined. If $\mathbf{B}_k = \mathbf{I}$, i.e. the simplest case of the descent direction being the gradient, the method is called gradient or *steepest descent*. If $\mathbf{B}_k = \mathbf{H}$, where $\mathbf{H}$ is the Hessian matrix, the method is the *Newton method*. The last family of methods are the *quasi-Newton* methods, where an approximation of the Hessian $\mathbf{B}_k \approx \mathbf{H}$ is used.

All of the mentioned methods have its advantages and disadvantages. The steepest descent is the easiest to implement, as it only needs the gradient of the function, but it has only linear convergence. The highest convergence can be ensured by the Newton method, which is quadratic in the best case. But it has the downside that it needs not only the gradient but also the complete Hessian. This can lead to a massive increase of computational complexity. The quasi-Newton methods are in between by having a superlinear convergence rate and only needing the gradient, which is also used for the approximation of the Hessian, as shown in the next section. For a deeper comparison of the methods and proofs to the mentioned convergence rates, the book of Nocedal and Wright [120] is recommended.

Quasi-Newton methods and BFGS update

Since quasi-Newton methods have a higher convergence as the steepest descent, without the need to explicitly calculate the Hessian like in the Newton method, they have become one of the most used methods. The basic idea for quasi-Newton methods is the approximation of the Hessian, i.e. $\mathbf{B}_k \approx \mathbf{H}$ using the *secant equation* (cf. Figure 4.2)

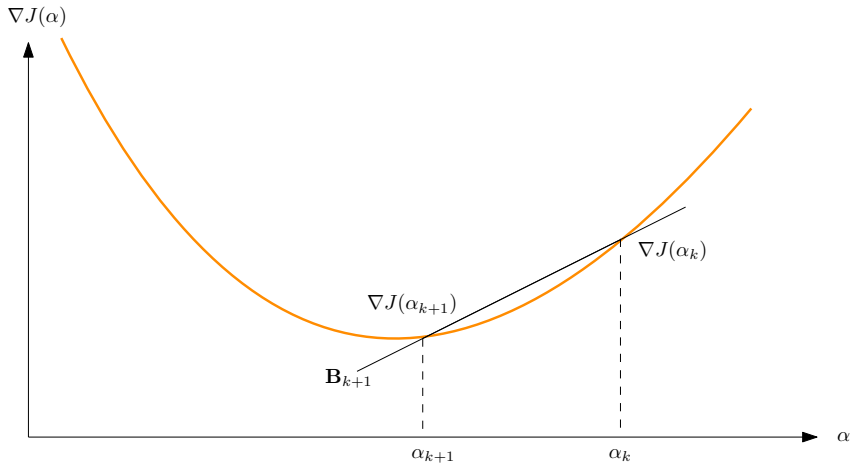


Figure 4.2: Illustration of the secant equation, used for the approximation of the Hessian in quasi-Newton methods. This illustration is based on the figure of De los Reyes [119].

$$\mathbf{B}_{k+1}\boldsymbol{s}_k = \boldsymbol{y}_k, \tag{4.15}$$

where

$$\boldsymbol{s}_k = \boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k, \tag{4.16}$$

$$\boldsymbol{y}_k = \nabla J(\boldsymbol{\alpha}_{k+1}) - \nabla J(\boldsymbol{\alpha}_k). \tag{4.17}$$

Under these quasi-Newton methods the *BFGS* method, which is named after its inventors Broyden, Fletcher, Goldfarb, and Shanno [123–127], has become the most popular [120]. The Hessian approximation can thereby be iteratively found solving a minimisation problem under a special weighted Frobenius norm, subject to certain constraints for the update, like being symmetric. With this the BFGS update reads

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^\top \mathbf{B}_k}{\boldsymbol{s}_k^\top \mathbf{B}_k \boldsymbol{s}_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^\top}{\boldsymbol{y}_k^\top \boldsymbol{s}_k}. \tag{4.18}$$

To ensure the descent direction of the BFGS update the initial approximation $\mathbf{B}_0$ needs to be positive definite and it must hold that $\boldsymbol{s}_k^\top \boldsymbol{y}_k > 0$. The latter can be seen when (4.15) is multiplied with $\boldsymbol{s}_k^\top$, from which follows that $\boldsymbol{s}_k^\top \mathbf{B}_{k+1} \boldsymbol{s}_k = \boldsymbol{s}_k^\top \boldsymbol{y}_k \overset{!}{\geq} 0$. If the Wolfe conditions (4.12) are used, then the condition $\boldsymbol{s}_k^\top \boldsymbol{y}_k > 0$ is implicitly given, which can be

seen in the following Lemma.

Lemma 4.2. *(Positive definiteness of the BFGS update) If the curvature condition* (4.12) *holds, then BFGS update is positive definite, i.e. condition* $\boldsymbol{s}_k^\top \boldsymbol{y}_k > 0$ *holds.*

Proof. Let $\nabla J(\boldsymbol{\alpha}_k) \stackrel{\text{def}}{=} \nabla J_k$ and use the curvature condition $\nabla J_{k+1}^\top \boldsymbol{p}_k \geq c_2 \nabla J_k^\top \boldsymbol{p}_k$ with $0 < c_2 < 1$ and $\boldsymbol{p}_k$ a descent direction, i.e. $\nabla J_k^\top \boldsymbol{p}_k < 0$. It follows

$$\begin{aligned}
\boldsymbol{y}_k^\top \boldsymbol{s}_k &= (\nabla J_{k+1} - \nabla J_k)^\top (\boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k) \\
&= (\nabla J_{k+1} - \nabla J_k)^\top \lambda_k \boldsymbol{p}_k \\
&\geq \lambda_k c_2 \nabla J_k^\top \boldsymbol{p}_k - \lambda_k \nabla J_k^\top \boldsymbol{p_k} \\
&= \lambda_k \nabla J_k^\top \boldsymbol{p}_k (c_2 - 1) \\
&> 0.
\end{aligned}$$

□

The search direction $\boldsymbol{p}_k = -\mathbf{B}_k^{-1} \nabla J_k$ uses the inverse of the approximative Hessian, but since this would be too expensive to be calculated in every step, the update of the inverse is used directly. This is derived using *Sherman–Morrison formula* as follows

$$\mathbf{B}_{k+1}^{-1} = (\mathbf{I} - \lambda_k \boldsymbol{s}_k \boldsymbol{y}_k^\top) \mathbf{B}_k^{-1} (\mathbf{I} - \lambda_k \boldsymbol{y}_k \boldsymbol{s}_k^\top) + \lambda_k \boldsymbol{s}_k \boldsymbol{s}_k^\top, \tag{4.19}$$

where $\lambda_k = \frac{1}{\boldsymbol{y}_k^\top \boldsymbol{s}_k}$ [128].

There have also been made improvements to the BFGS formula to cope with high memory requirements of large problems, called *limited-memory BFGS* or *L-BFGS* [129].

4.2.4 Algorithm

The following algorithm combines the line search method with Wolfe conditions and BFGS update. This algorithm will be used in the numerical experiments of this thesis. Typical start values are $\mathbf{B}_0 = \mathbf{I}$, $\lambda_0 = 1$, $c_1 = 10^{-4}$, $c_2 = 0.9$, $\rho_1 = 0.1$ and $\rho_2 = 2.1$.

Algorithm 2 Inexact line search with Wolfe conditions

Choose tol, $\mathbf{B}_0^{-1}, \boldsymbol{\alpha}_0, \lambda_0, c_1, c_2, \rho_1, \rho_2$
repeat
  Compute $\boldsymbol{p}_k = -\mathbf{B}_k^{-1}\nabla J_k$ ▷ Gradient based method
  Set $\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \lambda_k \boldsymbol{p}_k$ ▷ Line search
  while (False) **do**
    if $J_{k+1} \leq J_k + c_1 \nabla J_k^\top \boldsymbol{p}_k$ **then** ▷ Armijo condition
      $\lambda_k \leftarrow \lambda_k \rho_1$
      return False
    else if $\nabla J_{k+1}^\top \boldsymbol{p}_k \geq c_2 \nabla J_k^\top \boldsymbol{p}_k$ **then** ▷ Curvature condition
      $\lambda_k \leftarrow \lambda_k \rho_2$
      return False
    else
      return True
    end if
  end while
  Compute ∇J_{k+1}
  Set $\boldsymbol{s}_k = \boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k,\ \boldsymbol{y}_k = \nabla J_{k+1} - \nabla J_k$
  Compute $\mathbf{B}_{k+1}^{-1}$ using (4.19) ▷ Quasi-Newton method
  $k \leftarrow k + 1$
until $||\nabla J_k|| <$ tol
return

4.2.5 Numerical experiment

In order to test Algorithm 2 the non-convex Rosenbrock function [130] $f : \mathbb{R}^n \to \mathbb{R}$ is used, which is defined as [120, 131]

$$f(\boldsymbol{x}) = \sum_{i=1}^{n/2} \Big(\alpha\big(x_{2i} - x_{2i-1}^2\big) + \big(1 - x_{2i-1}\big)^2 \Big), \tag{4.20}$$

with $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})^\top$, $\alpha \in \mathbb{R}$ and $n \in \mathbb{N}$. The minimum of the Rosenbrock function is $f(\boldsymbol{x}^*) = 0$ for $\boldsymbol{x}^* = (1, 1, \ldots, 1)^\top$, and the gradient $\nabla f(\boldsymbol{x}) = (\frac{\partial f}{\partial x_0}, \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_{n-1}})^\top$ is given by

$$\frac{\partial f}{\partial x_{2i}} = 2\alpha\big(x_{2i} - x_{2i-1}^2\big), \tag{4.21}$$

$$\frac{\partial f}{\partial x_{2i-1}} = -4\alpha\big(x_{2i} - x_{2i-1}^2\big)x_{2i-1} - 2(1 - x_{2i-1}). \tag{4.22}$$

In the following the Rosenbrock function with $n = 10$, $\alpha = 100$, and start values $\boldsymbol{x}^0 = (1.0 \times 10^3, \ldots)^\top$ and $\boldsymbol{x}^1 = (1.5 \times 10^3, \ldots)^\top$ is investigated. Further, the results are tested with Armijo condition and Wolfe conditions, i.e. with and without the curvature

condition. The results are shown in Figure 4.3, and it can be seen that for start value $\boldsymbol{x}^0$ the optimisation method using Armijo condition and Wolfe conditions converges to the minimal value of $f^* = 0$, with Armijo condition even faster than with Wolfe conditions. For start value $\boldsymbol{x}^1$ only the optimisation method using Wolfe conditions converges to zero, while for the Armijo condition the optimiser gets stuck after 50 steps with a value of around 7031. These results indicate that the optimisation method works very well, and

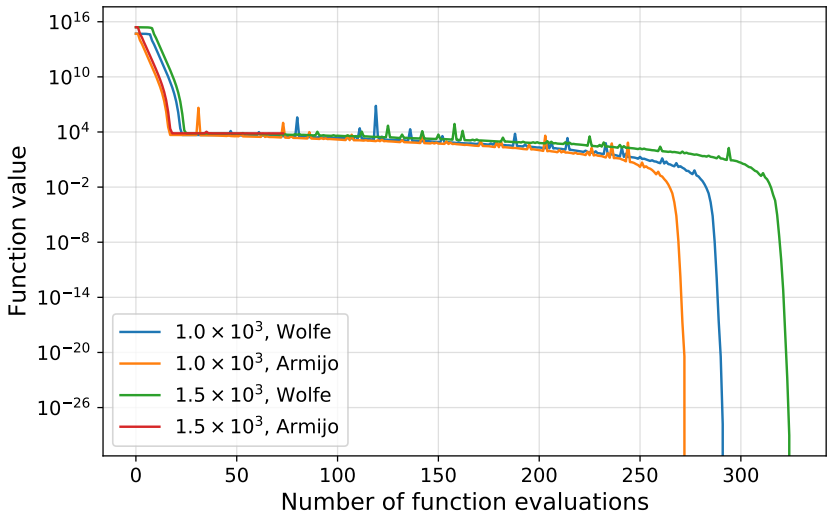


Figure 4.3: Minimising the Rosenbrock function using Algorithm 2 with (Wolfe) and without (Armijo) the curvature condition for different start values.

using the Wolfe conditions increases the stability of the method.

4.3 Gradient calculation

Unlike in the numerical example above, where the derivative of the Rosenbrock function is known, or can easily be derived, the computation of the gradient is one of the hardest parts of optimal control problems. Many different methods have evolved to tackle this challenge. The easiest and most basic maybe, being the *finite difference* method, which has its roots in Taylor's theorem. A more advanced method, in the sense of computer science at least, is *automatic differentiation*, also known as algorithmic differentiation, which manipulates the computer code in order to get the gradient in machine precision. The last method investigated and also used in this thesis, is the *adjoint approach* that uses the Lagrangian formalism and has a strong background in mathematical theory and is the most efficient method.

4.3.1 Finite difference method

The finite difference method belongs to algorithms of numerical differentiation, which approximately calculate the derivative from given function values. Let $J : \mathbb{R} \to \mathbb{R}$ be twice continuous differentiable with $\alpha \in \mathbb{R}$ and use a Taylor expansion of J at $\alpha + h$. This leads to

$$J'(\alpha) = \frac{J(\alpha + h) - J(\alpha)}{h} + \mathcal{O}(h), \tag{4.23}$$

also known as *forward difference* approximation. As can be seen, in order to get the derivative of J at point α, one needs to know the function value at $\alpha + h$, where h is some length. The error produced by this method is $\mathcal{O}(h)$, i.e. linear in h. This can be improved to quadratic error term, when combining forward and backward difference approximation, where the latter is basically the same as the first, only with an expansion at $J(\alpha - h)$. This then leads to the *central difference* approximation

$$J'(\alpha) = \frac{J(\alpha + h) - J(\alpha - h)}{2h} + \mathcal{O}(h^2). \tag{4.24}$$

Now the error produced is only $\mathcal{O}(h^2)$, but at the expense of now having to evaluate the function at two additional points.

The big advantage of finite differences are its very easy implementation. The disadvantages are performance and accuracy. If $\alpha \in \mathbb{R}^{1,000,000}$, which is common for topology optimisation problems, then one needs an extra two million function evaluations with the central difference approximation. For optimal control problems, this means solving a partial differential equation for this amount, which is not feasible. Another problem lies in numerical errors, which are produced if too small values for h are used, thus at some point round-off errors predominate [132], which is illustrated in Figure 4.4.

4.3.2 Automatic differentiation

The basis of automatic differentiation lies in the fact that in a computer program every function evaluation is a finite sequence of elementary operations [133]. Thus, all the computer needs to know are the *chain rule* and the rules to differentiate elementary operations. These are for example binary operations, such as addition, multiplication and unary operations like the sine function, and so on. This allows an almost exact evaluation of derivatives up to machine precision.

There are two main modes for automatic differentiation, called *forward* and *reverse* mode.

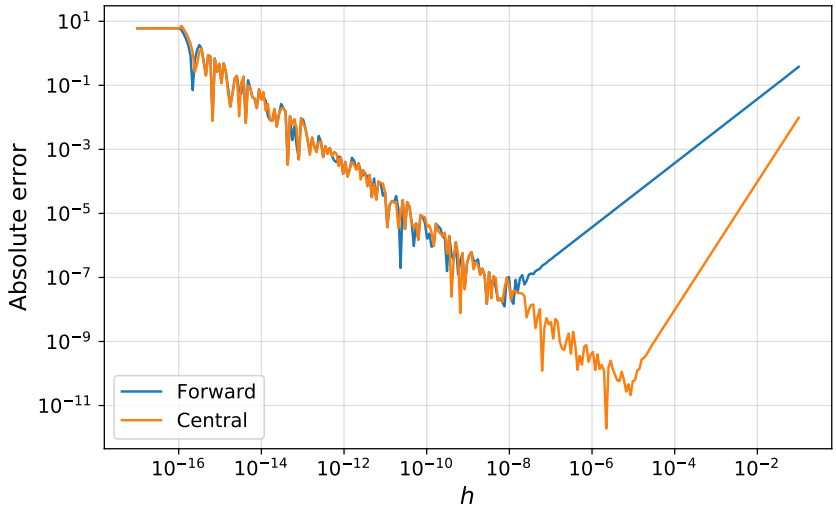


Figure 4.4: Absolute error from exact to approximated gradient for the function $J(\alpha) = \alpha^3 + \sin(\alpha)$ at $\alpha = 1.4$. Used are the forward and central difference approximation. The central starts with lower errors as the forward difference approximation, as expected, but at some point both experience round-off errors which then predominate.

The forward mode gives the derivative of each output with respect to the input variables and can run at the same time as the function evaluation. The reverse mode, as the name suggests, first computes all function evaluations, stores them and propagates the derivatives backward from a given output. Thus, the reverse mode is much more suitable for optimisation problems where the objective function maps from $\mathbb{R}^n$ to $\mathbb{R}$, as the gradient can be calculated by only one sweep. But at the same time the storage demands become infeasible for large n. There have been made many advances to overcome the issues, like *checkpointing strategies* [134], but this increases the complexity of the usage.

The implementation of automatic differentiation can be realised in different ways. One of them is the so-called *source transformation*, which takes the given program code and returns new code [135]. Another method uses the possibility of many programming languages, like for example C++, for *operator overloading*. This makes it possible to extend the elementary operations such as the exponential function or the multiplication so that not only the operation is executed, but also a new defined operation. In the case of automatic differentiation, this would then mean to calculate the analytical derivative of the operation together with the operation itself. If only the forward mode is used, which is feasible for smaller optimisation problems, the implementation is very straightforward. One such implementation for automatic differentiation in forward mode with C++, used in the lattice Boltzmann context, was presented by Krause et al. [136, 137].

A very related approach to the reverse mode automatic differentiation is the *adjoint approach* [134], which is based on a mathematically rigorous foundation.

4.3.3 Adjoint approach

The adjoint approach is the most efficient method, as the computational complexity stays constant [9], but it is the most complicated to derive and implement [118].

Here, first the general idea for finite dimensional case, then the formulation for function spaces are given, where similar to the automatic differentiation approach the chain rule is used. Let $J : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ and $\boldsymbol{R} : \mathbb{R}^m \times \mathbb{R}^n \to R^m$, and $f(\boldsymbol{\alpha})$ be the unique solution of $\boldsymbol{R}(f, \boldsymbol{\alpha}) = 0$.

Using the objective function $J(f(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ and the side condition $\boldsymbol{R}(f(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = 0$ and apply the chain rule

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial J}{\partial f}\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}} + \frac{\partial J}{\partial \boldsymbol{\alpha}}, \tag{4.25}$$

$$\frac{\mathrm{d}\boldsymbol{R}}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial \boldsymbol{R}}{\partial f}\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} = 0. \tag{4.26}$$

Thus, if $\frac{\partial \boldsymbol{R}}{\partial f}$ is invertible, then (4.26) can be solved for $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$ and used for the gradient of J

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}} = -\frac{\partial J}{\partial f}\frac{\partial \boldsymbol{R}}{\partial f}^{-1}\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} + \frac{\partial J}{\partial \boldsymbol{\alpha}}.$$

With this, the quite complicated operator $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$, which is the total derivative of the solution operator with respect to the control, can be substituted to relatively easy to derive partial derivatives.

Now, defining $\varphi^\top \overset{\text{def}}{=} -\frac{\partial J}{\partial f}\frac{\partial \boldsymbol{R}}{\partial f}^{-1}$ leads to

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}} = \varphi^\top \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} + \frac{\partial J}{\partial \boldsymbol{\alpha}}, \tag{4.27}$$

where φ is the solution of the *adjoint equation*

$$\frac{\partial \boldsymbol{R}}{\partial f}^\top \varphi = -\frac{\partial J}{\partial f}^\top. \tag{4.28}$$

This is a linear equation in φ, and the computationally expensive inversion of $\frac{\partial \boldsymbol{R}}{\partial f}$ is avoided. The function φ is called *Lagrangian multiplier*. This allows that for $\alpha \in \mathbb{R}^n$ only two equations need to be solved, regardless of the number of variables $n \in \mathbb{N}$. More complicated are derivation and discretisation, see Chapter 5. Another advantage is the similar structure of the adjoint equation and the side condition. This can lead to a similar discretisation and solving strategies, see for example Krause et al. [29].

Adjoint approach in function spaces

For the optimal control problem considered in this thesis, the concept of the adjoint approach is transferred to a function space setting (cf. [27]). For this, consider the optimal control problem (4.1), with $J : Y \times U \to \mathbb{R}$, $\boldsymbol{R} : Y \times U \to Z$, continuous Fréchet-differentiable operators between Banach spaces, with $W_{ad} \subset W \stackrel{\text{def}}{=} Y \times U$ nonempty and closed. Further, let $D_f \boldsymbol{R}\,(f(\alpha), \alpha) \in \mathcal{L}(Y, Z)$ be continuously invertible. Taking the derivative of $\boldsymbol{R}(f(\alpha), \alpha) = 0$ yields

$$D_y \boldsymbol{R}\,(y(\alpha), \alpha)\, f'(\alpha) + D_\alpha \boldsymbol{R}(f(\alpha), \alpha) = 0. \tag{4.29}$$

Using the duality pairing $\langle \cdot, \cdot \rangle_{X^*,X}$ for Banach space X and its dual space X^*, then the directional derivative of J in direction $h \in U$ reads

$$\begin{aligned} \big\langle J'(f(\alpha), \alpha), h \big\rangle_{U^*,U} &= \big\langle D_f J(f(\alpha), \alpha), f'(\alpha) h \big\rangle_{Y^*,Y} + \big\langle D_\alpha J(f(\alpha), \alpha), h \big\rangle_{U^*,U} && (4.30) \\ &= \big\langle f'(\alpha)^* D_f J(f(\alpha), \alpha), h \big\rangle_{U^*,U} + \big\langle D_\alpha J(f(\alpha), \alpha), h \big\rangle_{U^*,U}. && (4.31) \end{aligned}$$

Thus, as the direction h is arbitrary, the derivative of J reads

$$J'(f(\alpha), \alpha) = f'(\alpha)^* D_f J(f(\alpha), \alpha) + D_\alpha J(f(\alpha), \alpha). \tag{4.32}$$

Using (4.29) and (4.32), the derivative of J can be determined using

$$J'(f(\alpha), \alpha) = D_\alpha \boldsymbol{R}(f(\alpha), \alpha)^* \varphi(\alpha) + D_\alpha J(f(\alpha), \alpha), \tag{4.33}$$

where $\varphi = \varphi(\alpha) \in Z^*$ solves

$$D_f \boldsymbol{R}(f(\alpha), \alpha)^* \varphi = -D_f J(f(\alpha), \alpha). \tag{4.34}$$

With this the following *optimality system* can be derived.

Theorem 4.3 (Optimality system)**.** *Let* $\big(f(\bar{\alpha}), \bar{\alpha}\big) \in W_{ad}$ *be a local optimal solution of* (4.2) *and* $D_f \boldsymbol{R}(f(\bar{\alpha}), \bar{\alpha}) \in \mathcal{L}(Y, Z)$ *be a bijection, then there exists* $\varphi \in Z^*$ *with*

$$\boldsymbol{R}(f(\bar{\alpha}), \bar{\alpha}) = 0, \tag{4.35a}$$

$$D_f \boldsymbol{R}(f(\bar{\alpha}), \bar{\alpha})^* \varphi = -D_f J(f(\bar{\alpha}), \bar{\alpha}), \tag{4.35b}$$

$$D_\alpha \boldsymbol{R}(f(\bar{\alpha}), \bar{\alpha})^* \varphi = -D_\alpha J(f(\alpha), \bar{\alpha}). \tag{4.35c}$$

Proof. See for example [119]. □

Lagrangian formalism

Another and convenient way to derive the optimality system (4.35) is by defining a so-called *Lagrangian* $\mathrm{L} : Y \times U \times Z^* \to \mathbb{R}$ with

$$\mathrm{L}(f, \alpha, \varphi) = J(f, \alpha) + \langle \varphi, \boldsymbol{R}(f, \alpha) \rangle_{Z^*, Z} \,. \tag{4.36}$$

With it, the adjoint equation is to find $\varphi = \varphi(\alpha)$ such that

$$D_f \mathrm{L}(f(\alpha), \alpha, \varphi) = 0, \tag{4.37}$$

and thus using $\varphi(\alpha)$ from (4.37) the gradient of J reads

$$J'(f(\alpha), \alpha) = D_\alpha \mathrm{L}(f(\alpha), \alpha, \varphi(\alpha)). \tag{4.38}$$

For finite dimensional problems the biggest difference is that now the standard scalar product is used and the dual operator is now a transposed vector. Thus,

$$\mathrm{L}(f, \alpha, \varphi) = J\left(f(\alpha), \alpha\right) + \varphi^\top \boldsymbol{R}\left(f(\alpha), \alpha\right), \tag{4.39}$$

where the adjoint equation (4.28) is derived calculating $\frac{\partial \mathrm{L}}{\partial f} = 0$ and the gradient (4.27) by calculating $\frac{\partial \mathrm{L}}{\partial \alpha} = 0$.

Although it is very practical, the use of Lagrangian formalism must be well reasoned, as it must be ensured that not only the prerequisites are met, but also that the functional is differentiable and the adjoint operators exist in the correct spaces [118, 119]. Nonetheless, the method is very easy applicable and can give some understanding or hint to the adjoint equations. In Chapter 5 the *formal Lagrange technique* proposed by Tröltzsch [118] is used to derive the adjoint equations beforehand and then choose the right corresponding spaces.

Chapter 5

Combining measurement and simulation

Here, first the equations of the CFD-MRI method are presented, that is objective function and side condition. In order to solve the resulting optimisation problem the corresponding adjoint problem and optimality condition, and with it the gradient, is derived. Using a similar technique to the lattice Boltzmann method (cf. Section 2.3), the adjoint lattice Boltzmann method [136] is applied to discretise the adjoint problem. Then a projection method is proposed to link the control of the problem with the underlying porous media model of the side condition. Finally, the complete procedure for the CFD-MRI is presented.

The method and its results, with the exception of the derivation of the equations, have been published in [1–3, 30].

5.1 Objective function

The MRI data considered here is a time averaged image of a steady state flow denoted as $\boldsymbol{u}^* : \Omega \to \mathbb{R}^n$, where $\Omega \subset \mathbb{R}^d$ is the observed flow domain and $d, n \in \mathbb{N}$ the dimension of the spatial domain and velocity domain. The state, which will be the result of a fluid simulation, is denoted by $\boldsymbol{u} : \Omega \to \mathbb{R}^d$. In applications it is possible that MRI data are only available with one velocity component ($n = 1$), show only a section of the entire measurement experiment ($\Omega_{\text{MRI}} \subset \Omega$) or are only 2D spatially resolved ($d = 2$). All different combinations of information being available will be studied by numerical experiments in Chapters 6 to 8.

To measure the difference between data and state, an objective function $J : U_{ad} \to \mathbb{R}$ is sought. Here U_{ad} denotes the set of admissible controls $\alpha \in U_{ad}$ that determine the state $\boldsymbol{u}$. An intuitive approach of the measure is using

$$\int_\Omega |\boldsymbol{u} - \boldsymbol{u}^*| \, \mathrm{d}\boldsymbol{x}, \tag{5.1}$$

but the problem of this objective function is the L^1 structure, which leads to a convex but not necessarily differentiable functional [138]. A common choice is the use of a L^2 norm, due to the Hilbert space structure of L^2. Therefore the chosen objective function measures the L^2 distance from the measured flow field to the state

$$\frac{1}{2} \int_\Omega \left(\boldsymbol{u} - \boldsymbol{u}^*\right)^2 \mathrm{d}\boldsymbol{x}. \tag{5.2}$$

The factor of one-half is only used for simplicity's sake, such that it vanishes when considering the derivative, it does not change the optimisation process.

5.2 Side condition

Without any side condition, a solution to minimise the above objective function would be $\boldsymbol{u} \equiv \boldsymbol{u}^*$. This is neither wanted nor suitable, as the measurement $\boldsymbol{u}^*$ is contaminated by noise. Therefore the side condition should consider this and at best not allow any noise in the solution.

The MRI data is a measurement of fluid flow which can be described by conservation of mass and momentum (cf. Chapter 2). The noise in the data on the other hand does not fulfil these conservation laws, as it is mainly random fluctuation (cf. Chapter 3). Thus the natural choice of the side condition is to allow only those solutions $\boldsymbol{u}$ that satisfy these fundamental laws of fluid flow. Using this, noise can be eliminated and the procedure can be seen as a Navier–Stokes filter for MRI data, in the sense that it only allows those solutions that are close to the measured, noisy field but fulfil the macroscopic conservation laws.

As seen in Chapter 2 there are different possible choices to model fluid flow. Here we focus on the BGK-Boltzmann equation. Using the lattice Boltzmann method the BGK-Boltzmann equation leads to very efficient and highly parallel algorithms, which numerically solve the underlying fluid flow problem and thus ensure the conservation laws in the macroscopic setting [59].

5.3 CFD-MRI optimal control problem

The CFD-MRI method is formulated as topology optimisation problem which minimises the L^2 distance from given to computed fluid field, where the latter is based on the porous media BGK-Boltzmann equation.

Considering a domain $\Omega \subset \mathbb{R}^3$, time interval $I = [0, T]$ for some $T > 0$, and velocity space $\Xi \subseteq \mathbb{R}^3$, the CFD-MRI problem is written as

$$\min\ J(f, \alpha) \stackrel{\text{def}}{=} \frac{1}{2} \int_\Omega \left(\boldsymbol{u}_f(T, \boldsymbol{x}) - \boldsymbol{u}^*(\boldsymbol{x}) \right)^2 \mathrm{d}\boldsymbol{x} \tag{5.3}$$

subject to

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f = -\frac{1}{\tau}(f - f_\alpha^{\text{eq}}) \qquad \text{in } \Sigma \stackrel{\text{def}}{=} I \times \Omega \times \Xi, \tag{5.4a}$$

$$f(t, \boldsymbol{x}, \boldsymbol{\xi}) = f(t, \boldsymbol{x}, -\boldsymbol{\xi}) \qquad \text{on } \Gamma \stackrel{\text{def}}{=} I \times \partial\Omega \times \Xi, \tag{5.4b}$$

$$f(0, \boldsymbol{x}, \boldsymbol{\xi}) = f_0(\boldsymbol{x}, \boldsymbol{\xi}) \qquad \text{in } \Omega \times \Xi, \tag{5.4c}$$

with the Maxwellian distribution function

$$f_\alpha^{\text{eq}}(t, \boldsymbol{x}, \boldsymbol{\xi}) = \frac{\rho_f(t, \boldsymbol{x})}{(2\pi RT)^{3/2}} e^{-\frac{1}{2RT}|\boldsymbol{\xi} - \alpha(t,\boldsymbol{x})\boldsymbol{u}_f(t,\boldsymbol{x})|^2}. \tag{5.5}$$

The macroscopic quantities of density and velocity can be derived by

$$\rho_f(t, \boldsymbol{x}) = \int_\Xi f(t, \boldsymbol{x}, \boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi}, \qquad \boldsymbol{u}_f(t, \boldsymbol{x}) = \rho_f^{-1} \int_\Xi \boldsymbol{\xi} f(t, \boldsymbol{x}, \boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi}. \tag{5.6}$$

The control $\alpha : I \times \Omega \to \mathbb{R}$ thereby should satisfy

$$0 \leq \alpha(t, \boldsymbol{x}) \leq 1 \quad \text{a.e. in } I \times \Omega, \tag{5.7}$$

and enters the Maxwellian distribution function to be able to model porous media, as shown in Section 2.3.3. The system (5.3)–(5.7) is the optimal control problem for CFD-MRI and used in this thesis.

5.4 Adjoint CFD-MRI problem

In order to solve the optimal control problem (5.3)–(5.7), the adjoint approach presented in Section 4.3.3, employing the formal Lagrange technique [118], is used.

Let $\alpha \in U_{ad} \stackrel{\text{def}}{=} \{\alpha \in L^2(I \times \Omega) :\ 0 \leq \alpha(t, \boldsymbol{x}) \leq 1 \text{ a.e. in } \Sigma\}$ and $Y \stackrel{\text{def}}{=} \{f \in L^2(\Sigma) : \partial_t f \in L^2(\Sigma),\ \nabla_{\boldsymbol{x}} f, \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f \in L^2(\Sigma),\ f, \boldsymbol{\xi} f \in L^1(\Xi),\ \boldsymbol{\xi} \in \Xi\}$. Further, it is assumed that for any $\alpha \in U_{ad}$ there exists a unique solution $f(\alpha) \in Y$ with $\boldsymbol{R}(f, \alpha) = 0$, where the latter is the residual form of (5.4).

Definition 5.1. *For problem* (5.3)–(5.4) *the Lagrangian* $\mathrm{L} : Y \times U_{ad} \times Y \to \mathbb{R}$ *is defined as*

$$\begin{aligned} \mathrm{L}(f, \alpha, \varphi) = \frac{1}{2} \int_\Omega \big(\boldsymbol{u}_f(T, \boldsymbol{x}) - \boldsymbol{u}^*(\boldsymbol{x})\big)^2 \,\mathrm{d}\boldsymbol{x} \\ + \int_\Sigma \varphi_1(t, \boldsymbol{x}, \boldsymbol{\xi}) \Big(\partial_t f + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f + \frac{1}{\tau}(f - f_\alpha^{eq})\Big)(t, \boldsymbol{x}, \boldsymbol{\xi}) \,\mathrm{d}t \,\mathrm{d}\boldsymbol{x} \,\mathrm{d}\boldsymbol{\xi} \\ + \int_\Gamma \varphi_2(t, \boldsymbol{x}, \boldsymbol{\xi}) \Big(f(t, \boldsymbol{x}, \boldsymbol{\xi}) - f(t, \boldsymbol{x}, -\boldsymbol{\xi})\Big) \,\mathrm{d}t \,\mathrm{d}s(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{\xi}, \end{aligned} \tag{5.8}$$

with Lagrangian multiplier $\varphi = (\varphi_1, \varphi_2)$.

From the Lagrangian principle it follows that for an optimal solution $(\bar{f}, \bar{\alpha})$ it holds that $D_f L(\bar{f}, \bar{\alpha}, \varphi)(f - \bar{f}) \geq 0$ for every sufficiently smooth f with $f(0, \cdot, \cdot) = f_0$. Redefining $f \stackrel{\text{def}}{=} f - \bar{f}$ it follows $D_f L(\bar{f}, \bar{\alpha}, \varphi) f = 0$ for every f with $f(0, \cdot, \cdot) = 0$ (cf. [118]).

5.4.1 Derivation of the adjoint problem

The adjoint problem is derived by setting the partial derivative of the Lagrangian with respect to the state to zero, i.e. $D_f \mathrm{L}(\bar{f}, \bar{\alpha}, \varphi) f = 0$, see Section 4.3.3.

Using the Lagrangian (5.8) it follows for $h \in Y$

$$\begin{aligned} D_f \mathrm{L}(f, \alpha, \varphi) h = D_f J(f, \alpha, \varphi) h \\ + \int_\Sigma \big(D_f(\partial_t f) h + D_f(\boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f) h\big) \,\mathrm{d}t \,\mathrm{d}\boldsymbol{x} \,\mathrm{d}\boldsymbol{\xi} \\ + \int_\Sigma \frac{1}{\tau} \big(D_f f h - D_f f_\alpha^{\mathrm{eq}} h\big) \,\mathrm{d}t \,\mathrm{d}\boldsymbol{x} \,\mathrm{d}\boldsymbol{\xi} \\ + \int_\Gamma D_f \big(f(t, \boldsymbol{x}, \boldsymbol{\xi}) - f(t, \boldsymbol{x}, -\boldsymbol{\xi})\big) h \,\mathrm{d}t \,\mathrm{d}s(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{\xi}. \end{aligned} \tag{5.9}$$

In the following the needed partial derivatives with respect to the state of certain quantities, like the moments, the equilibrium distribution and the objective functions are derived.

Proposition 5.1 (Linear derivatives)**.** *For $h \in Y$ the derivatives of the linear terms of the Lagrangian* (5.9) *are*

$$D_f f h = h, \tag{5.10}$$

$$D_f \rho_f h = \int_\Xi h \,\mathrm{d}\boldsymbol{\xi}, \tag{5.11}$$

$$D_f (\rho_f \boldsymbol{u}_f) h = \int_\Xi \boldsymbol{\xi} h \,\mathrm{d}\boldsymbol{\xi}, \tag{5.12}$$

$$D_f \big(\partial_t f + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f\big) h = \partial_t h + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} h. \tag{5.13}$$

Proof. The moments ρ_f and $\rho_f \boldsymbol{u}_f$ are linear operations and thus the derivative is the operator itself (cf. Chapter 4). To see this, let $\rho(f) \overset{\text{def}}{=} \rho_f = \int_\Xi f \,\mathrm{d}\boldsymbol{\xi}$ be the density moment depending on the distribution function f. Using the definition of the directional derivative (4.4) yields

$$\begin{aligned} D_f \rho_f h &= \lim_{t \to 0} \frac{1}{t} \big(\rho(f + th) - \rho(f)\big) \\ &= \lim_{t \to 0} \frac{1}{t} \left(\int_\Xi f + th \,\mathrm{d}\boldsymbol{\xi} - \int_\Xi f \,\mathrm{d}\boldsymbol{\xi} \right) \\ &= \lim_{t \to 0} \frac{1}{t} \left(\int_\Xi th \,\mathrm{d}\boldsymbol{\xi} \right) \\ &= \int_\Xi h \,\mathrm{d}\boldsymbol{\xi}. \end{aligned}$$

Similar, using $\rho \boldsymbol{u}(f) \overset{\text{def}}{=} \rho_f \boldsymbol{u}_f = \int_\Xi \boldsymbol{\xi} f \,\mathrm{d}\boldsymbol{\xi}$, i.e. the momentum moment with respect to f and thus

$$\begin{aligned} D_f (\rho_f \boldsymbol{u}_f) h &= \lim_{t \to 0} \frac{1}{t} \big(\rho \boldsymbol{u}(f + th) - \rho \boldsymbol{u}(f)\big) \\ &= \lim_{t \to 0} \frac{1}{t} \left(\int_\Xi \boldsymbol{\xi}(f + th) \,\mathrm{d}\boldsymbol{\xi} - \int_\Xi \boldsymbol{\xi} f \,\mathrm{d}\boldsymbol{\xi} \right) \\ &= \int_\Xi \boldsymbol{\xi} h \,\mathrm{d}\boldsymbol{\xi}. \end{aligned}$$

The same holds for the time and convective derivative, i.e.

$$\begin{aligned} &D_f \big(\partial_t f + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f\big) h = \partial_t h + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} h, \\ &D_f f h = h. \end{aligned}$$

□

With this the derivatives of the non-linear terms can now be computed, which are summarised in the following Lemma.

Lemma 5.1 (Non-linear derivatives)**.** *For $h \in Y$ the derivatives of the non-linear terms of the Lagrangian* (5.9) *are*

$$
\begin{aligned}
D_f \boldsymbol{u}_f h &= \int_\Xi \frac{\boldsymbol{\xi} - \boldsymbol{u}_f}{\rho_f} h \,\mathrm{d}\boldsymbol{\xi}, \\
D_f J h &= \int_\Omega \int_\Xi \frac{(\boldsymbol{u}_f - \boldsymbol{u}^*)^\top (\boldsymbol{\xi} - \boldsymbol{u}_f)}{\rho_f} h \,\mathrm{d}\boldsymbol{\xi} \,\mathrm{d}\boldsymbol{x}, \\
D_f f_\alpha^{eq} h &= \int_\Xi \frac{f_\alpha^{eq}}{\rho_f} \left(1 + \frac{\alpha}{RT} (\boldsymbol{\xi} - \alpha \boldsymbol{u}_f)^\top (\hat{\boldsymbol{\xi}} - \boldsymbol{u}_f)\right) h \,\mathrm{d}\hat{\boldsymbol{\xi}}.
\end{aligned}
$$

Proof. For the velocity it holds $\boldsymbol{u}_f = \frac{\rho_f \boldsymbol{u}_f}{\rho_f}$ and thus

$$
\begin{aligned}
D_f \boldsymbol{u}_f h &= D_f \left(\frac{\rho_f \boldsymbol{u}_f}{\rho_f}\right) h \\
&= \frac{D_f(\rho_f \boldsymbol{u}_f) h \rho_f - \rho_f \boldsymbol{u}_f D_f \rho_f h}{\rho_f^2} \\
&= \int_\Xi \frac{\boldsymbol{\xi} - \boldsymbol{u}_f}{\rho_f} h \,\mathrm{d}\boldsymbol{\xi}.
\end{aligned}
$$

With the derivative of the velocity above it follows

$$
\begin{aligned}
D_f J h &= D_f \left(\frac{1}{2} \int_\Omega (\boldsymbol{u}_f - \boldsymbol{u}^*)^2 \,\mathrm{d}\boldsymbol{x}\right) h \\
&= \int_\Omega (\boldsymbol{u}_f - \boldsymbol{u}^*)^\top D_f \boldsymbol{u}_f h \,\mathrm{d}\boldsymbol{x} \\
&= \int_\Omega \int_\Xi \frac{(\boldsymbol{u}_f - \boldsymbol{u}^*)^\top (\boldsymbol{\xi} - \boldsymbol{u}_f)}{\rho_f} h \,\mathrm{d}\boldsymbol{\xi} \,\mathrm{d}\boldsymbol{x}.
\end{aligned}
$$

For the equilibrium distribution function the derivative of the density and velocity are needed and therefore

$$
\begin{aligned}
D_f f_\alpha^{\mathrm{eq}} h &= D_f \left(\frac{\rho_f}{(2\pi RT)^{3/2}} e^{-\frac{1}{2RT}|\boldsymbol{\xi} - \alpha \boldsymbol{u}_f|^2}\right) h \\
&= D_f \rho_f h \frac{f_\alpha^{\mathrm{eq}}}{\rho_f} + f_\alpha^{\mathrm{eq}} D_f \left(-\frac{1}{2RT} |\boldsymbol{\xi} - \alpha \boldsymbol{u}_f|^2\right) h \\
&= D_f \rho_f h \frac{f_\alpha^{\mathrm{eq}}}{\rho_f} + f_\alpha^{\mathrm{eq}} \frac{1}{RT} (\boldsymbol{\xi} - \alpha \boldsymbol{u}_f)^\top \alpha D_f \boldsymbol{u}_f h \\
&= \int_\Xi \frac{f_\alpha^{\mathrm{eq}}}{\rho_f} \left(1 + \frac{\alpha}{RT} (\boldsymbol{\xi} - \alpha \boldsymbol{u}_f)^\top (\hat{\boldsymbol{\xi}} - \boldsymbol{u}_f)\right) h(\hat{\boldsymbol{\xi}}) \,\mathrm{d}\hat{\boldsymbol{\xi}}.
\end{aligned}
$$

□

Using the derivatives presented in Lemma 5.1 it follows

$$\begin{aligned} D_f \mathrm{L}(\bar{f}, \bar{\alpha}, \varphi) f = &\int_\Omega \int_\Xi \frac{\left(\boldsymbol{u}_{\bar{f}}(T) - \boldsymbol{u}^*\right)^\top \left(\boldsymbol{\xi} - \boldsymbol{u}_{\bar{f}}(T)\right)}{\rho_{\bar{f}}(T)} f(T) \\ &+ \int_\Sigma \left(\varphi_1 \partial_t f + \varphi_1 (\boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f) + \frac{1}{\tau} \varphi_1 f \right) \\ &- \frac{1}{\tau} \int_\Sigma \underbrace{\int_\Xi \varphi_1(\hat{\boldsymbol{\xi}}) \frac{\bar{f}^{\mathrm{eq}}_{\bar{\alpha}}(\hat{\boldsymbol{\xi}})}{\rho_{\bar{f}}} \left(1 + \frac{\alpha}{RT} (\hat{\boldsymbol{\xi}} - \alpha \boldsymbol{u}_{\bar{f}})^\top (\boldsymbol{\xi} - \boldsymbol{u}_{\bar{f}})\right) \mathrm{d}\hat{\boldsymbol{\xi}}}_{\stackrel{\mathrm{def}}{=} \varphi^{\mathrm{eq}}_\alpha(\boldsymbol{\xi})} f(\boldsymbol{\xi}) \\ &+ \int_\Gamma \varphi_2 \big(f(t, \boldsymbol{x}, \boldsymbol{\xi}) - f(t, \boldsymbol{x}, -\boldsymbol{\xi}) \big). \end{aligned} \tag{5.14}$$

Here the differentials were omitted for better readability. Using integration by parts and Greens formula, the time and the convective differential operator can be transferred to the Lagrangian multiplier. The other operations are linear and therefore do not need to be modified. With a collection of matching integration domains this then leads to

$$\begin{aligned} D_f \mathrm{L}(\bar{f}, \bar{\alpha}, \varphi) f = &\int_\Omega \int_\Xi \frac{\left(\boldsymbol{u}_{\bar{f}}(T) - \boldsymbol{u}^*\right)^\top \left(\boldsymbol{\xi} - \boldsymbol{u}_{\bar{f}}(T)\right)}{\rho_{\bar{f}}(T)} f(T) \\ &+ \int_\Omega \int_\Xi \varphi_1(T) f(T) \\ &+ \int_\Sigma \left(-\partial_t \varphi_1 - (\boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} \varphi_1) + \frac{1}{\tau} (\varphi_1 - \varphi^{\mathrm{eq}}_\alpha) \right) f \\ &+ \int_\Gamma \Big(\boldsymbol{\xi} \cdot \boldsymbol{n} \varphi_1 f + \varphi_2 \big(f(t, \boldsymbol{x}, \boldsymbol{\xi}) - f(t, \boldsymbol{x}, -\boldsymbol{\xi}) \big) \Big). \end{aligned} \tag{5.15}$$

For an optimal solution it follows that (5.15) must be equal to zero for every f. Now assuming that $f \in C_0^\infty(\Sigma)$, it follows $f|_\Gamma = 0$ on and $f(T) = 0$, and thus

$$\int_\Sigma \left(-\partial_t \varphi_1 - \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} \varphi_1 + \frac{1}{\tau} (\varphi_1 - \varphi^{\mathrm{eq}}_\alpha) \right) f = 0 \quad \forall f.$$

Assuming $C_0^\infty(\Sigma)$ is dense in $L^2(\Sigma)$ it follows

$$-\partial_t \varphi_1 - \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} \varphi_1 = -\frac{1}{\tau} (\varphi_1 - \varphi^{\mathrm{eq}}_\alpha) \quad \text{in } I \times \Omega \times \Xi.$$

Now, let $f(T) \neq 0$, this leads to

$$\int_\Omega \int_\Xi \left(\varphi_1(T) + \frac{\left(\boldsymbol{u}_{\bar{f}}(T) - \boldsymbol{u}^*\right)^\top \left(\boldsymbol{\xi} - \boldsymbol{u}_{\bar{f}}(T)\right)}{\rho_{\bar{f}}(T)} \right) f(T) = 0 \quad \forall f.$$

As $f(T)$ is arbitrary, and assuming the possible values are dense in $L^2(\Omega \times \Xi)$ (cf. [118]),

it follows

$$\varphi_1(T) = -\frac{\big(\boldsymbol{u}_{\bar{f}}(T) - \boldsymbol{u}^*\big)^\top \big(\boldsymbol{\xi} - \boldsymbol{u}_{\bar{f}}(T)\big)}{\rho_{\bar{f}}(T)} \quad \text{in } \Omega \times \Xi.$$

With the same arguments and for $f|_\Gamma \neq 0$ the remaining term leads to (cf. Krause [136]) $\varphi_1 = \varphi_2 \overset{\text{def}}{=} \varphi$ and

$$\varphi(t, \boldsymbol{x}, \boldsymbol{\xi}) = \varphi(t, \boldsymbol{x}, -\boldsymbol{\xi}) \quad \text{on } I \times \partial\Omega \times \Xi.$$

Now the adjoint system of the optimal control problem for CFD-MRI is derived. This is summarised in the next theorem.

Theorem 5.1 (Adjoint problem)**.** *The adjoint problem of* (5.3)–(5.4) *is*

$$-\frac{\partial \varphi}{\partial t} - \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} \varphi = -\frac{1}{\tau}(\varphi - \varphi^{eq}) \qquad \text{in } I \times \Omega \times \Xi, \tag{5.16a}$$

$$\varphi(t, \boldsymbol{x}, \boldsymbol{\xi}) = \varphi(t, \boldsymbol{x}, -\boldsymbol{\xi}) \qquad \text{on } I \times \partial\Omega \times \Xi, \tag{5.16b}$$

$$\varphi(T) = -\frac{\big(\boldsymbol{u}_f(T) - \boldsymbol{u}^*\big)^\top \big(\boldsymbol{\xi} - \boldsymbol{u}_f(T)\big)}{\rho_f(T)} \qquad \text{in } \Omega \times \Xi, \tag{5.16c}$$

with $\varphi_\alpha^{eq}(\boldsymbol{\xi}) = \rho_f^{-1} \int_\Xi \varphi(\hat{\boldsymbol{\xi}}) f_\alpha^{eq}(\hat{\boldsymbol{\xi}}) \big(1 + \frac{\alpha}{RT}(\hat{\boldsymbol{\xi}} - \alpha \boldsymbol{u}_f)^\top (\boldsymbol{\xi} - \boldsymbol{u}_f)\big)\, \mathrm{d}\hat{\boldsymbol{\xi}}$.

Note, if the objective function (5.3) would be a function of time, i.e. without a steady state solution, the derivative of the objective function would not be a part of the initial condition, but rather the adjoint equation (5.16a), with the initial condition (5.16c) then being $\varphi(T) = 0$, see Krause [136].

Now that the adjoint system is derived, the derivation of the optimality condition comes next. This is not only one of the necessary conditions for optimal solutions, but also needed to calculate the gradient in order to solve the optimal control problem, as demonstrated in Chapter 4.

5.4.2 Derivation of the optimality condition

The optimality condition can be derived using the Lagrangian method, see Section 4.3.3, and reads $D_\alpha \mathrm{L}(\bar{f}, \bar{\alpha}, \varphi)(\alpha - \bar{\alpha}) \geq 0$. Again, from now on the differentials are omitted for better readability.

Since only the equilibrium distribution function depends on the control, it follows

$$\begin{aligned} D_\alpha \mathrm{L}(f,\alpha,\varphi)h &= D_\alpha\Big(\int_\Sigma \varphi\big(\partial_t f + \boldsymbol{\xi}\cdot\partial_{\boldsymbol{x}} f + \frac{1}{\tau}(f - f_\alpha^{\mathrm{eq}})\big)\Big)h \\ &= -\frac{1}{\tau}\int_\Sigma \varphi D_\alpha f_\alpha^{\mathrm{eq}} h, \end{aligned}$$

where the derivative of the equilibrium distribution function with respect to the control is

$$\begin{aligned} D_\alpha f_\alpha^{\mathrm{eq}} h &= D_\alpha\left(\frac{\rho_f}{(2\pi RT)^{3/2}} e^{-\frac{1}{2RT}|\boldsymbol{\xi}-\alpha\boldsymbol{u}_f|^2}\right) h \\ &= -f_\alpha^{\mathrm{eq}} D_\alpha\left(\frac{1}{2RT}|\boldsymbol{\xi}-\alpha\boldsymbol{u}_f|^2\right) h \\ &= \frac{1}{RT} f_\alpha^{\mathrm{eq}}\big(\boldsymbol{\xi}-\alpha\boldsymbol{u}_f\big)^\top \boldsymbol{u}_f h. \end{aligned}$$

Theorem 5.2 (Optimality condition)**.** *Using the above the optimality condition for system* (5.3)–(5.4) *can be derived as*

$$D_\alpha \mathrm{L}(\bar{f},\bar{\alpha},\varphi)(\alpha-\bar{\alpha}) = -\frac{1}{\tau RT}\Big(\int_\Sigma \varphi \bar{f}_{\bar{\alpha}}^{eq}\big(\boldsymbol{\xi}-\bar{\alpha}\boldsymbol{u}_{\bar{f}}\big)^\top \boldsymbol{u}_{\bar{f}}\Big)(\alpha-\bar{\alpha}) \geq 0. \tag{5.17}$$

With this the gradient of the objective function can be written as

$$J'(f,\alpha) = -\frac{1}{\tau RT}\int_\Xi \varphi(\boldsymbol{\xi}) f_\alpha^{eq}(\boldsymbol{\xi})\big(\boldsymbol{\xi}-\alpha\boldsymbol{u}_f\big)^\top \boldsymbol{u}_f \,\mathrm{d}\boldsymbol{\xi}. \tag{5.18}$$

Note that only the dependency on velocity is shown here, for easier readability.

5.5 Discretisation

In order to solve the adjoint system numerically, a discretisation scheme must be applied, very similar to Section 2.3. This discretisation strategy is called *adjoint lattice Boltzmann method* (ALBM) and proposed by Krause et al. [136, 137].

As seen in the previous section, the gradient of the objective function (5.3) with side condition (5.4) can be calculated by

$$J'(f,\alpha) = -\frac{1}{\tau RT}\int_\Xi \varphi(\boldsymbol{\xi}) f_\alpha^{\mathrm{eq}}(\boldsymbol{\xi})\big(\boldsymbol{\xi}-\alpha\boldsymbol{u}_f\big)^\top \boldsymbol{u}_f \,\mathrm{d}\boldsymbol{\xi},$$

where φ is the solution of the adjoint problem

$$
\begin{aligned}
-\frac{\partial \varphi}{\partial t} - \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} \varphi &= -\frac{1}{\tau}(\varphi - \varphi^{\mathrm{eq}}) && \text{in } I \times \Omega \times \Xi, \\
\varphi(t, \boldsymbol{x}, \boldsymbol{\xi}) &= \varphi(t, \boldsymbol{x}, -\boldsymbol{\xi}) && \text{on } I \times \partial\Omega \times \Xi, \\
\varphi(T) &= -\frac{\big(\boldsymbol{u}_f(T) - \boldsymbol{u}^*\big)^\top \big(\boldsymbol{\xi} - \boldsymbol{u}_f(T)\big)}{\rho_f(T)} && \text{in } \Omega \times \Xi,
\end{aligned}
$$

with adjoint equilibrium distribution function $\varphi_\alpha^{\mathrm{eq}}(\boldsymbol{\xi}) = \rho_f^{-1} \int_\Xi \varphi(\hat{\boldsymbol{\xi}}) f_\alpha^{\mathrm{eq}}(\hat{\boldsymbol{\xi}}) \big(1 + \frac{\alpha}{RT}(\hat{\boldsymbol{\xi}} - \alpha \boldsymbol{u}_f)^\top (\boldsymbol{\xi} - \boldsymbol{u}_f)\big)\, \mathrm{d}\hat{\boldsymbol{\xi}}$.

Like for the BGK-Boltzmann equation a discrete space $I_h \times \Omega_h \times Q$ has to be chosen, denoted as $DdQq$ with d the dimension and q the number of discrete velocities in $Q \subset \Xi$. For the ALBM $D3Q19$ is used here, like for the LBM in Section 2.3. Note that by using the first-optimise-then-discretise approach the discretisation model could also be used differently to the forward problem [136].

In a very similar manner to the LBM in Section 2.3 an iterative algorithm can be derived, which is executed step by step but for decreasing $t \in I_h$. With $\varphi_i(t, \boldsymbol{x}) \overset{\mathrm{def}}{=} \frac{w_i}{\omega(\boldsymbol{c}_i)} \varphi(t, \boldsymbol{x}, \boldsymbol{c}_i)$ the adjoint lattice Boltzmann equation thus reads

$$
\varphi_i(t - \triangle t, \boldsymbol{x} - \boldsymbol{c}_i \triangle t) = \varphi_i(t, \boldsymbol{x}) - \frac{\triangle t}{\tau} \big(\varphi_i(t, \boldsymbol{x}) - \varphi_i^{\mathrm{eq}}(t, \boldsymbol{x})\big), \tag{5.20}
$$

for all $\boldsymbol{x} \in \Omega_h$ and every $i = 0, 1, ..., q-1$ and with adjoint initial condition for $t = T$

$$
\varphi_i(T) = -\frac{\big(\boldsymbol{u}_f(T) - \boldsymbol{u}^*\big)^\top \big(\boldsymbol{c}_i - \boldsymbol{u}_f(T)\big)}{\rho_f(T)}.
$$

Thereby the moments, $\boldsymbol{u}_f$, and ρ_f, are computed by the same quadrature rule as for the forward problem, i.e. the LBE, and the microscopic velocity $\boldsymbol{\xi}$ is replaced by the quadrature nodes $\boldsymbol{c}_i$ for $i \in \{0, 1, \ldots, q-1\}$. For the adjoint equilibrium distribution function $\varphi_\alpha^{\mathrm{eq}}$ the exponential function of f_α^{eq} is employed to apply Gauss–Hermite quadrature, where the remaining terms are combined to a polynomial function. This polynomial function has the same degree as used for the moments of the LBE, thus the quadrature can be expected to be of the same quality. The discrete adjoint equilibrium distribution function $\varphi_i^{\mathrm{eq}}(t, \boldsymbol{x}) \overset{\mathrm{def}}{=} \varphi^{\mathrm{eq}}(t, \boldsymbol{x}, \boldsymbol{c}_i)$ can therefore be derived as

$$
\varphi_i^{\mathrm{eq}} = \rho_f^{-1} \sum_{j=0}^{q-1} \varphi_j f_{j,\alpha}^{\mathrm{eq}} \left(1 + \frac{\alpha}{c_s^2} (\boldsymbol{c}_j - \alpha \boldsymbol{u}_f)^\top (\boldsymbol{c}_i - \boldsymbol{u}_f)\right), \tag{5.21}
$$

with the quadrature weights collected in $f_{j,\alpha}^{\mathrm{eq}}$ as before and $RT \overset{\mathrm{def}}{=} c_s^2$. The same can be

applied for the gradient for which the discrete version is thus given by

$$J_h' = -\frac{1}{\tau c_s^2} \sum_{i=0}^{q-1} \varphi_i f_{i,\alpha}^{\mathrm{eq}} \big(\boldsymbol{c}_i - \alpha \boldsymbol{u}_f\big)^\top \boldsymbol{u}_f. \tag{5.22}$$

Because the boundary conditions are all of Dirichlet type the bounce back condition is employed which has the same formulation as for the LBE [136].

By comparing the LBE (2.74) with the adjoint equation (5.20), the similarity can be seen immediately. The biggest differences are the temporal progression and the initial condition. However, the structure is identical, like the locality of the collision, which leads to the fact that the same highly parallelisable implementation can be used [139]. For a more detailed description of an efficient implementation [30] is recommended.

5.6 Projection

In topology optimisation problems of fluid flows, the porous medium is often introduced by a force

$$\boldsymbol{f} = -\beta(\rho)\boldsymbol{u}, \tag{5.23}$$

where $\beta(\rho) \in \mathbb{R}$ is modelled as inverse permeability, or porosity, and a function of the density ρ. For this, most often interpolation functions are used [10, 16, 20, 140, 141], where one caveat is the presence of many user-defined parameters which are not physically motivated.

These aforementioned works consider the extreme cases, i.e. pure solid or fluid nodes, but the intermediate values can also be linked to permeability of the domain. Here, a projection method is proposed which maps the control to the feasible set and at the same time using the structure of the porous media BGK-Boltzmann equation (5.4). Consequently, the unbounded optimisation method shown in Algorithm 2 can be used.

The results of this section have been published in [2].

5.6.1 Grid dependency of control

In the model used here, the parameter that controls the topology in the domain is porosity $d_h \in [0, 1]$ [12], but as shown in Section 2.3.3 the porosity $d_h \in [0, 1]$ is related to the

permeability K by

$$\begin{aligned} d_h &= 1 - \frac{\triangle x^2 \nu \tau}{K} \\ &\stackrel{\text{def}}{=} 1 - \frac{G_h}{K}, \end{aligned} \tag{5.24}$$

and by this, the porosity is highly grid dependent (cf. Figure 5.1). Thus using the porosity

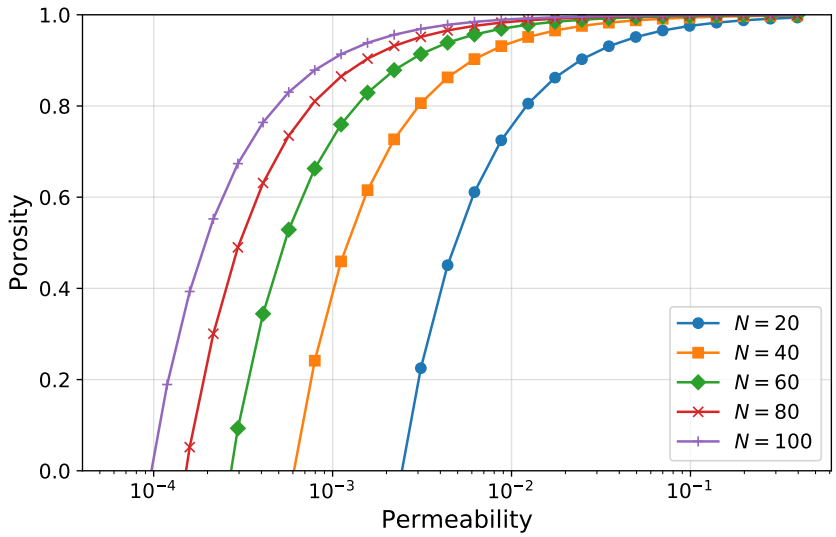


Figure 5.1: Connection of permeability and porosity for the same setup as in Figure 5.2. The same porosity will lead to different permeability values for different resolutions $N = 1/\triangle x$, i.e. number of nodes per side length of 1 m.

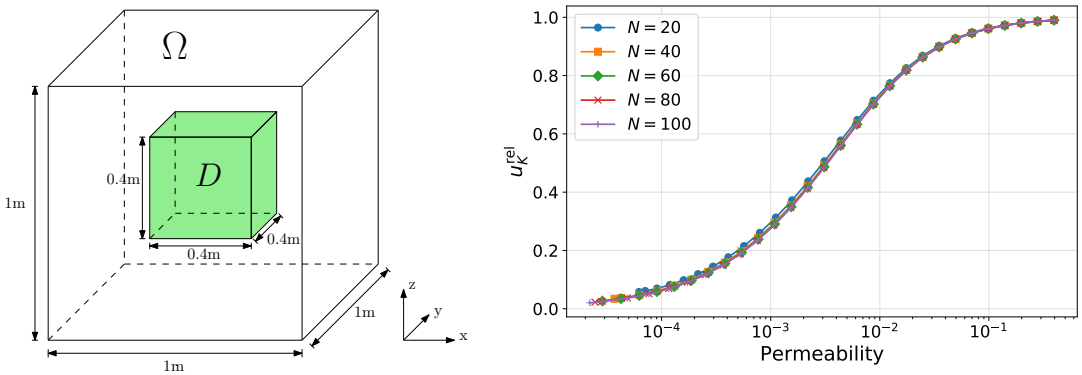
as control can lead to unwanted parameter tuning. Porosity is only a design parameter in porous media simulations, however, the permeability K is the grid-independent parameter which determines the flow behaviour, as shown in the following test case.

5.6.2 Grid independency of permeability

In order to show the grid independence of the permeability a test case is constructed, where the flow through a porous domain is analysed. The domain $\Omega \subset \mathbb{R}^3$ is a cubic domain from $(0,0,0)^\top$m to $(1,1,1)^\top$m, with a smaller porous domain $D \subset \Omega$ which has a given permeability K and extents from $(0.3,0.3,0.3)^\top$m to $(0.7,0.7,0.7)^\top$m. For the setup see Figure 5.2. The boundary conditions are set to constant flow with 1 m/s at the front $y = 0$ and back $y = 1$. Bounce back is set on the remaining walls. The flow has a kinematic viscosity of $0.1\,\text{m}^2/\text{s}$. To analyse the flow for different permeabilities K, the relative velocity, averaged in domain D, denoted as $\boldsymbol{u}_K^{\text{rel}}$ is considered, i.e.

$$\boldsymbol{u}_K^{\text{rel}} \stackrel{\text{def}}{=} \frac{||\boldsymbol{u}_K||_{L^2(D)}}{||\boldsymbol{u}_\infty||_{L^2(D)}}. \tag{5.25}$$

Here $\boldsymbol{u}_K$ is the velocity in domain D with given permeability K, and therefore $\boldsymbol{u}_\infty$ is the flow for pure fluid domains, i.e. $d_h = 1$.



(a) Test case

(b) Permeability grid independence

Figure 5.2: Test case for the grid independence study for the permeability. Left, the setup is shown. Right, the results of the test case. The flow is analysed using the relative velocity averaged inside the porous domain D for different resolutions N.

The test case shows that permeability determines the behaviour of the flow, and is grid independent at that. The resulting plot (Figure 5.2a), also shows a symmetric behaviour in logarithmic representation. Therefore, coupling the control to the permeability indicates to be a more suitable choice, as it directly gives the sensitivity of the flow with respect to the control.

5.6.3 Constructing a projection

By (5.24) for the porosity being between 0 and 1 the permeability K has to be in the interval $[G_h, \infty)$. Thus to couple the control with the permeability a sub-projection is needed for which $p : \mathbb{R} \to [0, \infty)$ with $K \mapsto p(\alpha) + G_h$ holds. Therefore the porosity value depending on control can be written as

$$\begin{aligned} d_h(\alpha) &= 1 - \frac{G_h}{p(\alpha) + G_h} \\ &= \frac{p(\alpha)}{p(\alpha) + G_h}. \end{aligned} \tag{5.26}$$

For the CFD-MRI method the following projection, $P \in C^\infty(\mathbb{R})$, is proposed

$$P : \begin{cases} \mathbb{R} \to (0, 1) \\ \alpha \mapsto P\alpha \overset{\text{def}}{=} \dfrac{e^\alpha}{e^\alpha + G_h}, \end{cases} \tag{5.27}$$

where $G_h = \triangle x^2 \nu_h \tau_h$.

Using this projection the control is now directly mapped to permeability and thus grid-independent, see Figure 5.3.

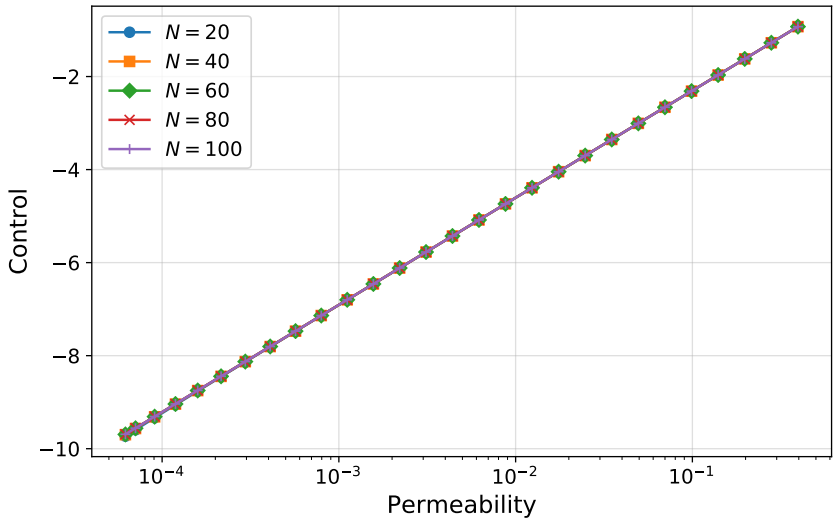


Figure 5.3: Connection of permeability and control.

Using the proposed projection an unbounded optimisation method can be used, and the control now has a structure which is consistent with the porous media model. Note, for $G_h = 1$ this projection is the sigmoid function often used as activation function in artificial neural networks [142]. Since the sigmoid function and the proposed projection are equivalent up to translation, the sigmoid function can be used if the start value is chosen on the basis of permeability. This can be much easier since the grid term G_h does not have to be computed and used.

5.7 CFD-MRI procedure

With the results of Section 2.3 and Section 5.5 and using a line search method with Wolfe conditions (Algorithm 2) the procedure of CFD-MRI is presented as follows.

First an MRI measurement of the experimental setup is carried out, e.g. the flow around or through an object in a pipe. The external geometry, here e.g. the pipe dimension, as well as the properties of the fluid and the used inflow velocity, are then used for the simulation. The internal structure is not known at first, but is found automatically by the CFD-MRI method. Note that the same applies to the MRI measurement, where only the object, the properties of the used fluid, and the inflow velocity is known. The interior only becomes visible through the measurement itself. So the same knowledge is available

for the setup of measurement and simulation.

For CFD-MRI, first a starting value for the optimisation is selected, which is sensitive with regard to the underlying porous media model. Next, the simulation is carried out and the resulting flow field is compared with the MRI data using the objective function. The result of this evaluation then leads to a change in the starting value for a better approximation. This is done using the line search method, which requires the gradient of the objective function. To calculate the gradient, the adjoint problem is solved, which receives the results of the simulation as input data. This loop of simulation, adjoint problem and line search is repeated until the objective function is minimised. The procedure is summarised in Algorithm 3.

Algorithm 3 CFD-MRI

Get MRI data
Set outer-geometry, inflow velocity, and viscosity for the simulation based on the experimental setup
repeat
    Run simulation using a lattice Boltzmann equation (2.74)
    Calculate difference from simulation and measurement by (5.3)
    Solve an adjoint lattice Boltzmann equation (5.20) to get the gradient by (5.22)
    Update the control using line search by Algorithm 2
    Project control to porosity by (5.27)
until Simulation matches MRI data

Chapter 6

Application of CFD-MRI to a flow MRI measurement

In this chapter the proof of concept of the CFD-MRI method applied to MRI data is presented. The aim is to demonstrate the feasibility of the method and to conduct preliminary qualitative analyses.

This chapter is a revised version of [1] and summarises its main results.

6.1 MRI experimental setup

The experimental setup is a flow of isopropanol around a cylindrical object in a pipe using a chromatographic pump with minimal pulsation to ensure a stationary flow field. The experiment was performed by Pro^{2}NMR of Karlsruhe Institute of Technology (KIT) on a 200 MHz super wide bore MRI instrument by Bruker, Biospin GmbH equipped with an Avancce HDIII console and gradients up to 1 T/m for the imaging and flow encoding. To greatly reduce the measurement time, only $2D$ spatially resolved flow images were acquired in a measurement time of several minutes. The MRI data consists of 100×255 data points, containing only the velocity component in the direction of the flow, see Figure 6.1.

6.2 Simulation setup

The simulation is set up to match the MRI experiment. The domain is a pipe with length of 25.5×10^{-3} m and radius of 4.0×10^{-3} m. The fluid has a density of $\rho = 768\,\mathrm{kg/m^2}$

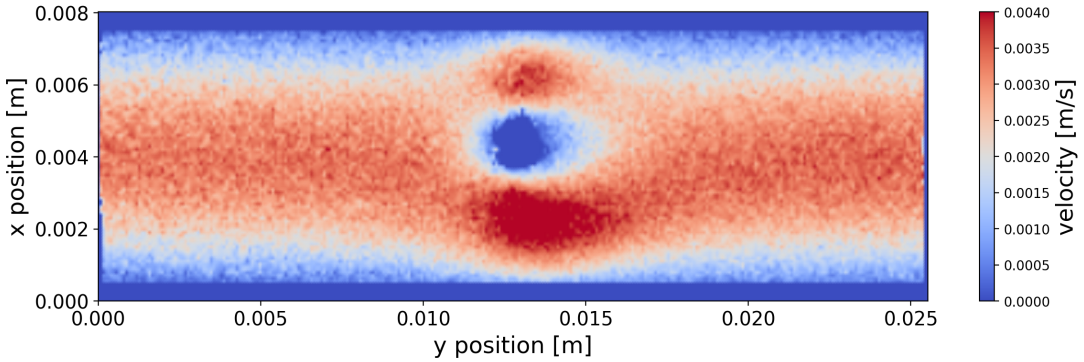


Figure 6.1: MRI data of a flow around a cylindrical object acquired by Pro^{2}NMR at KIT.

and kinematic viscosity of $\nu = 2.798 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$ corresponding to the properties of isopropanol. At the inflow a Poiseuille profile with maximum velocity of $3.6 \times 10^{-3}\,\mathrm{m/s}$, and at the outflow a constant pressure, i.e. free outflow, is chosen. The remaining boundaries are set to bounce back boundary conditions, i.e. no slip. The simulation domain is discretised by $100 \times 255 \times 100$ nodes, matching the MRI data in the plane. The relaxation time for LBM is chosen to be $\tau = 0.57$.

For the optimisation the controls are set to $\alpha = -19$, which relates to a permeability of around $K = 5.68 \times 10^{-9}\,\mathrm{m}^2$, in a cylindrical domain of length $4.0 \times 10^{-3}\,\mathrm{m}$ and radius of $2.0 \times 10^{-3}\,\mathrm{m}$ starting at $y = 11.0 \times 10^{-3}\,\mathrm{m}$, see Figure 6.2, where the object is assumed, and set to be fluid nodes else.

The MRI data being only $2D$ spatially resolved, and having only one velocity component pose the two biggest challenges for the CFD-MRI method. For this, the objective domain Ω_J is reduced to the location of the MRI data, i.e. the x–y plane at $z = 4.0 \times 10^{-3}\,\mathrm{m}$. And the objective function is reduced to only account for the $y-$velocity of the flow, i.e.

$$J(\alpha) = \frac{1}{2}||u_y(\alpha) - u^*||^2_{L^2(\Omega_J)}. \tag{6.1}$$

Here u_y denotes the $y-$component of the simulated velocity, and u^* the corresponding MRI data. The data is provided on the lattice nodes using trilinear interpolation. Additionally, the implementation was extended to be able to handle data with arbitrary location and degrees of freedom.

6.3 CFD-MRI results

The value of the objective function, $J = \frac{1}{2}||u_y - u^*||^2_{L^2(\Omega_J)}$, starts at 8.63×10^{-2} and is reduced to 1.56×10^{-2} after only five optimisation steps and converges to 0.63×10^{-2} at

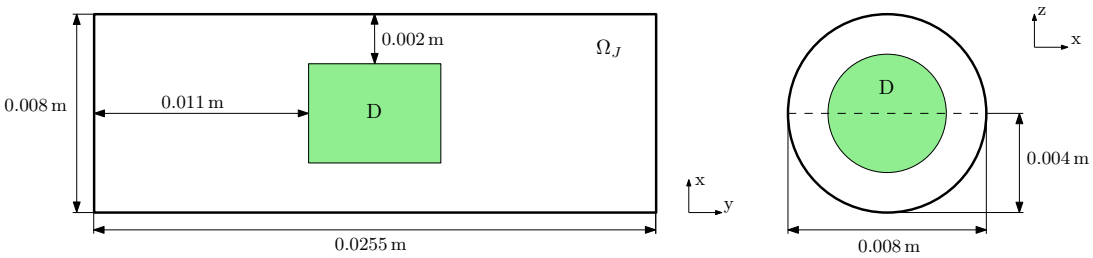


Figure 6.2: The simulation domain is a pipe with radius of 4.0×10^{-3} m and length of 25.5×10^{-3} m. The MRI data is located in the x–y plane (left) and indicated by the dashed line (right). The objective domain Ω_J matches the MRI data. The design domain D, shown in green, is placed where the object is assumed and starts with a permeability of around $K = 6.1 \times 10^{-9}\,\mathrm{m}^2$.

only twenty-five steps, see Figure 6.3. When comparing the relative error from simulated

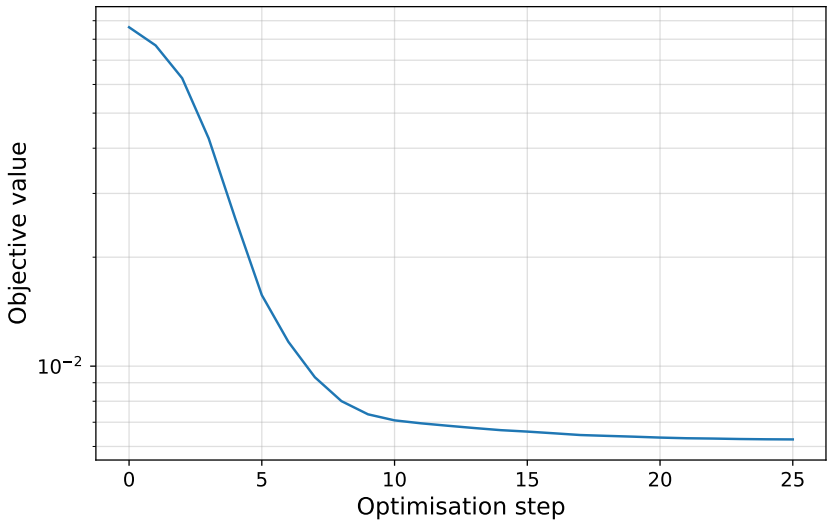


Figure 6.3: Value of the objective function, $\frac{1}{2}||u_y - u^*||^2_{L^2(\Omega_J)}$, of MRI data u^* to CFD-MRI result u_y over the optimisation steps.

to measured velocity, i.e.

$$\frac{||u_y - u^*||_{L^2(\Omega_J)}}{||u^*||_{L^2(\Omega_J)}}, \tag{6.2}$$

the error starts at 41.50%, is reduced to 22.47% after five steps and ends with 10.87%, which is a fourfold reduction in relative error. Comparing MRI data with the CFD-MRI result in Figure 6.4 there is a clear reduction of the measurement noise while at the same time finding the object in the flow very well. In Figure 6.5 a plot in the middle of the domain in the y–z plane is shown with simulated and measured velocity. It can be seen that the measurement noise is significantly reduced and thus the 10.87% error could indicate the measurement noise. Also, the CFD-MRI result does fulfil conservation laws as it is the solution of the underlying porous media BGK-Boltzmann equation, which is not the case for the MRI data. The result of the object identification is shown in Figure 6.6,

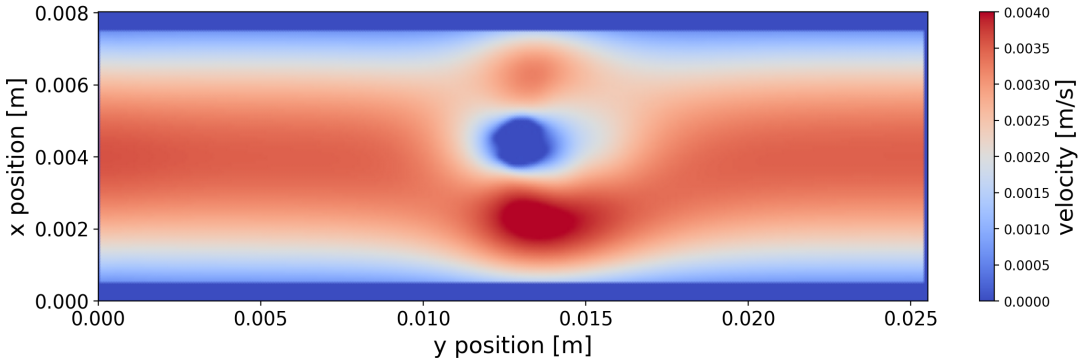


Figure 6.4: Result of the CFD-MRI method applied to the MRI data shown in Figure 6.1.

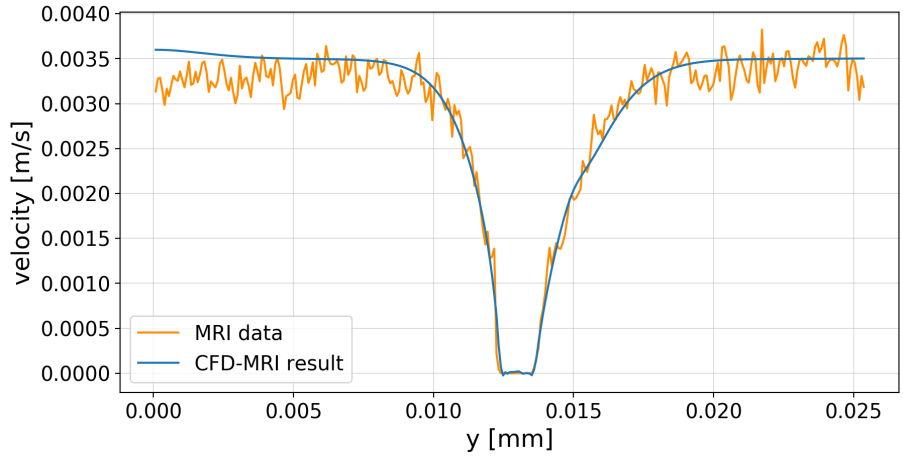
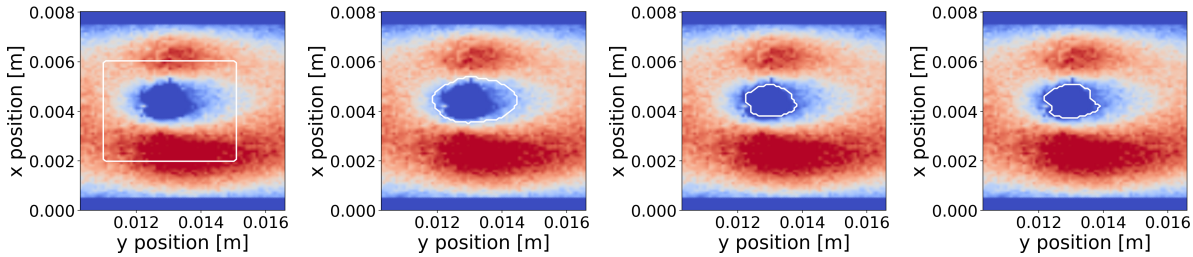


Figure 6.5: Plot over line in the middle of the long side of the pipe with MRI data and CFD-MRI result.

with the MRI data in the background and the contour of the object shown as white line. After only two steps the basic area of the object is found, which gets more precise after ten steps. The outline of the object in the MRI data is found very precisely after only twenty-five steps.



(a) Step 0 (b) Step 2 (c) Step 10 (d) Step 25

Figure 6.6: Visualisation of the object identification after 0, 2, 10 and 25 optimisation steps. In the background the MRI data is shown, the white line is a contour plot of the object found by the CFD-MRI method.

6.4 Conclusion

The results of this chapter indicate that the method is able to identify objects in the flow and at the same time reducing the measurement noise. In order to analyse the method and validate these results, in Chapter 7 the object identification capabilities are further studied using simulation data. In Chapter 8 the method is analysed with regard to noise reduction using synthetic MRI data.

Chapter 7

Object identification in fluid flow

In this chapter the proposed method is analysed with respect to its object identification capabilities. The goal is to identify objects present in the flow based only on the velocity data. Here, two different objects are considered, one with a simple, symmetric and one with a more complex, asymmetric geometry. Further, the available data will be gradually reduced in order to analyse the quality of the identification based on the given data.

This is a revised version of [2]. In addition, the Stanford bunny object [143] and the analysis using binary classification are presented.

7.1 Test case

The test cases consist of a cubical domain Ω of 1 m side length, filled with a fluid with a density of $\rho = 1$ kg/m^3 and kinematic viscosity of $\nu = 0.1$ m^2/s. Boundary conditions are set to Dirichlet velocity boundaries with 1 m/s in $y-$direction. The to be identified solid objects are placed in the middle of the domain. The domain is discretised with a resolution of $N = 104$ nodes per 1 m length, resulting in a space discretisation of $\triangle x = 9.62 \times 10^{-3}$ m and thus $1,124,864$ lattice nodes. The relaxation time was chosen as $\tau = 0.8$.

To be able to analyse the method, first a simulation is carried out with a to be identified solid object, $O \subset \Omega$, in the middle of the domain Ω. The lattice-porosities in the domain are set to solid, i.e. $d_h(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in O$, and to fluid in the remaining domain, i.e. $d_h(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in \Omega \backslash O$. The resulting flow field $\boldsymbol{u}_f$ is then saved as $\boldsymbol{u}^*$ to be the artificial MRI data. The CFD-MRI method is then applied to the artificial data as shown in Algorithm 3. For this the object is discarded and a start value α_0 is chosen. Using the proposed projection

P in (5.27) this control maps to a lattice-porosity value of $d_h(\boldsymbol{x}) = P\alpha_0$ for $\boldsymbol{x} \in D$. For the remaining domain the lattice-porosity is set $d_h(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in \Omega \setminus D$. A detailed description of the test cases are found in Figures 7.1 and 7.11.

The procedure is summarised in Algorithm 4.

Algorithm 4 Object identification test case

Create artificial data
  Simulate fluid flow with $d_h(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in O$ and $d_h(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in \Omega \setminus O$
  Save the resulting flow data as the solution flow field $\boldsymbol{u}^*$
Apply CFD-MRI
  Choose start value α_0 and set $d_h(\boldsymbol{x}) = P\alpha_0$ for $\boldsymbol{x} \in D$ and $d_h(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in \Omega \setminus D$
  Solve optimisation problem using Algorithm 3 where the MRI data is replaced by $\boldsymbol{u}^*$

7.2 Classification

Here the binary classification [144] is used to evaluate the performance of object identification. As the result of the optimisation method is a continuous control, a threshold is chosen for which the lattice-porosities are either considered solid or fluid [145]. Considering the predicted versus the actual class of a node, following four outcomes can be determined, see Table 7.1.

True positive	(TP)	Identified as solid is actual solid node
True negative	(TN)	Identified as fluid is actual fluid node
False positive	(FP)	Identified as solid is actual fluid node
False negative	(FN)	Identified as fluid is actual solid node

Table 7.1: Binary classification outcomes.

For the evaluation of the performance of the classification there exists several measures [146]. *Accuracy* is the percentage of right prediction from all outcomes

$$\frac{TP + TN}{TP + TN + FP + FN}. \tag{7.1}$$

Recall measures how many of the actual solid nodes were correctly identified

$$\frac{TP}{TP + FN}. \tag{7.2}$$

Precision is the measure of correct from all predicted solid nodes

$$\frac{TP}{TP + FP}. \tag{7.3}$$

The *Jaccard index* is the measure of intersection over union, i.e. for two sets A and B the Jaccard index is defined as $J(A,B) \stackrel{\text{def}}{=} \frac{|A \cap B|}{|A \cup B|}$ [147, 148]. For binary classification this equals

$$\frac{TP}{TP + FP + FN}, \tag{7.4}$$

and describes the similarity between the identified and the actual object [149, 150]. Note that all measures above have a range of $[0, 1]$.

7.3 Simple object

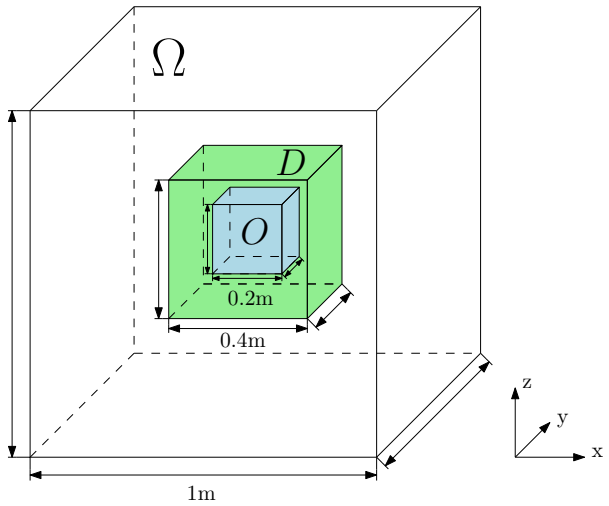


Figure 7.1: Setup for the cube identification test case. The object to be identified is a cube with side length of 0.2 m placed inside the middle of the domain Ω. The design domain D, where the porosity will be changed, is a cube with side length of 0.4 m.

In this test case (cf. Figure 7.1) the to be identified object $O \subset D \subset \Omega$ is a small cube with side length of 0.2 m located in the middle of the domain Ω. The design domain $D \subset \Omega$, where the lattice-porosity is changed, is a larger cube with side length of 0.4 m.

The objective value, $J(\alpha) = \frac{1}{2}||\boldsymbol{u}_f(\alpha) - \boldsymbol{u}^*||^2_{L^2(\Omega)}$, decreases clearly, as seen in Figure 7.2. Starting from $J = 7.23 \times 10^{-3}$ in the first step, the objective value is reduced by a factor of 635 after only ten steps. And after sixty steps the objective function has value of $J = 7.97 \times 10^{-8}$ and is thus reduced by the order of 10^5 in total. For a more intuitive comparison the relative error from simulated to given flow field

$$\frac{||\boldsymbol{u}_f - \boldsymbol{u}^*||_{L^2(\Omega)}}{||\boldsymbol{u}^*||_{L^2(\Omega)}}, \tag{7.5}$$

is used. Here, the error starts at 12% and decreases to 0.4% after only ten steps, and

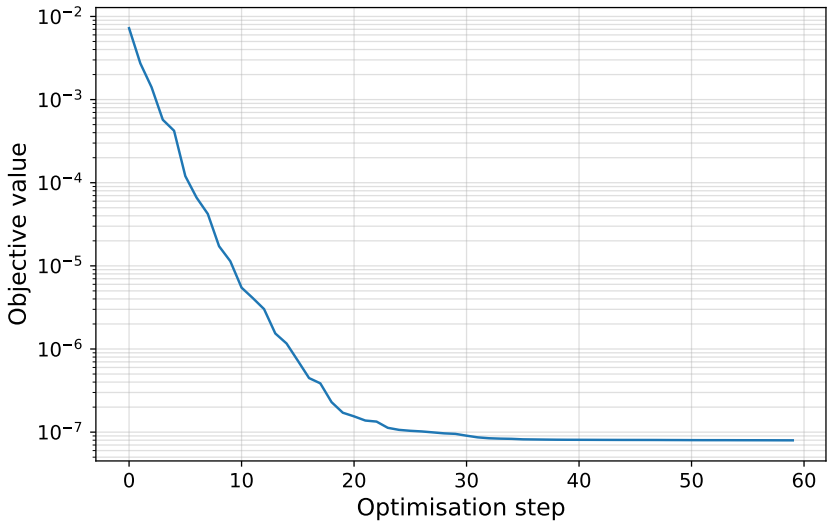


Figure 7.2: Plot of the objective value, $\frac{1}{2}||\boldsymbol{u}_f - \boldsymbol{u}^*||^2_{L^2(\Omega)}$, over the optimisation steps.

decreases even further to 0.04% after sixty steps. This shows the clear matching of simulated to given flow field.

In order to evaluate the object identification performance, first the L^2 error in the true object domain, i.e. $||d_h||_{L^2(O)}$, is measured. Since the object was initialised with $d_h^* = 0$ for the data generation, it can be assumed that if the identification is successful, the norm within the object domain, should tend towards zero. As shown in Figure 7.3, the porosity

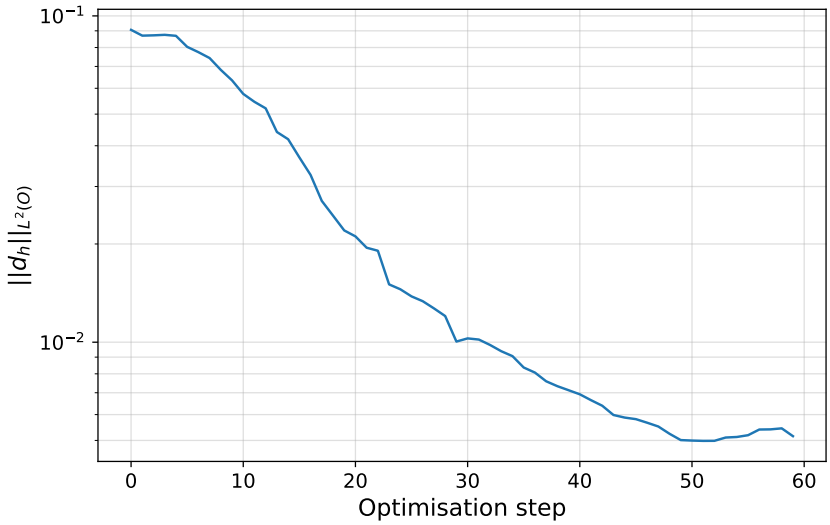
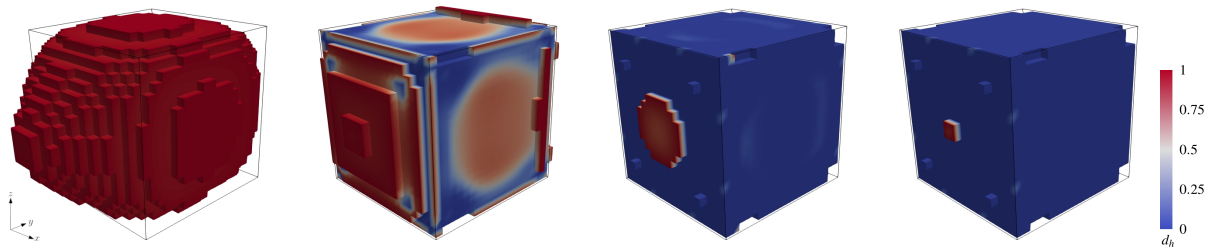


Figure 7.3: Plot of the porosity norm inside the object domain, $||d_h||_{L^2(O)}$, over the optimisation steps. Note, the true object has a norm of $||d_h^*||_{L^2(O)} = 0$.

norm inside the object domain does indeed decrease but not as fast as the objective value.

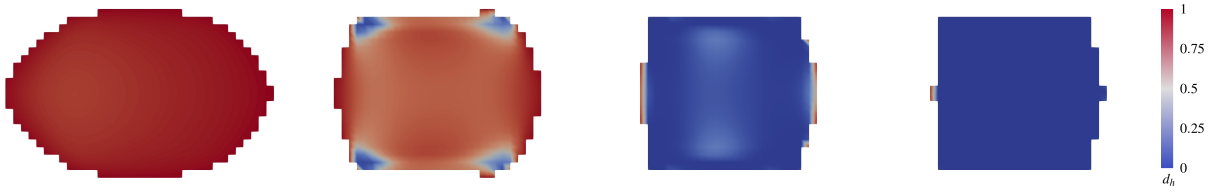
The visual representation of the object for different optimisation steps is shown in Figure 7.4, where the outline of the true object is marked in black. For the graphical visualisation of the numerical results of porosity threshold value of $d_h = 0.99$ is chosen, meaning



(a) 5 steps (b) 10 steps (c) 25 steps (d) 50 steps

Figure 7.4: Visualisation of the cube identification after 5, 10, 25 and 50 optimisation steps. The original cube is marked by a black outline. The colour represents lattice-porosities for $d_h \leq 0.99$.

every porosity value greater than the threshold is considered fluid and not shown. The cube can be clearly identified in only ten optimisation steps, with a precise identification after fifty steps with very low porosity values, indicating a very clear separation between solid and fluid points. In order to visually evaluate the result in the inside of the domain Figure 7.5 shows a slice in the y–z plane. It can be seen that even the inside of the cube is getting more solid not only the outer shell.



(a) 5 steps (b) 10 steps (c) 25 steps (d) 50 steps

Figure 7.5: Results of the cube identification inside the object, shown in a slice of the cube in the y–z plane after 5, 10, 25 and 50 optimisation steps. The colour represents lattice-porosities for $d_h \leq 0.99$.

For a more detailed analysis of how well the object identification works, the above mentioned classification measures are used. First, a threshold has to be chosen, which treats a given porosity value as either solid or fluid. Figure 7.6 shows the classification measures with respect to the threshold from $d_h = 0.001$ to $d_h = 0.999$. Table 7.2 shows the measures for selected thresholds in detail. The maximum Jaccard index is found at a

Threshold	*0.1*	*0.5*	*0.9*	*0.99*	∅
Accuracy	**0.9998**	**0.9998**	**0.9998**	**0.9998**	0.9998
Recall	0.9909	0.997	0.9978	**0.9987**	0.9952
Precision	**0.978**	0.9778	0.9774	0.9752	0.9776
Jaccard index	0.9694	0.9749	**0.9753**	0.9740	0.9730

Table 7.2: Classification measures for the object identification with different thresholds of lattice-porosities for which the resulting object is considered to be solid. And the mean of 100 threshold values from $d_h = 0.001$ to $d_h = 0.999$.

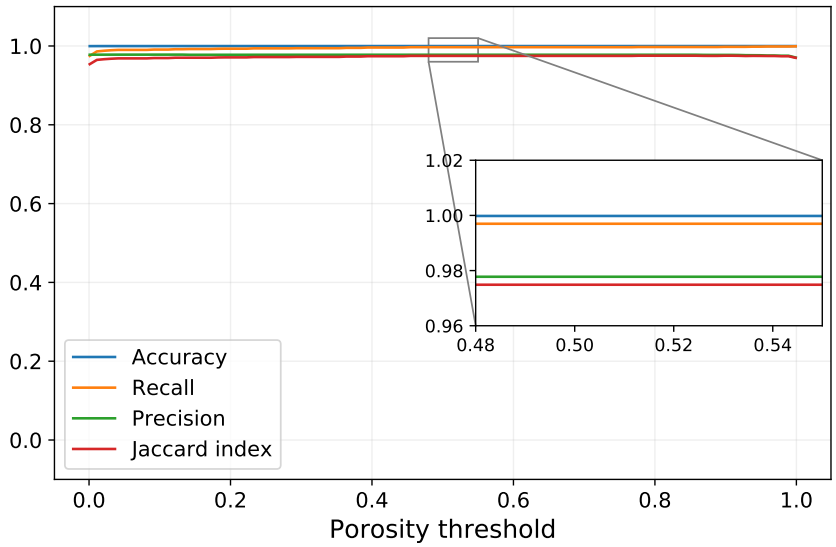
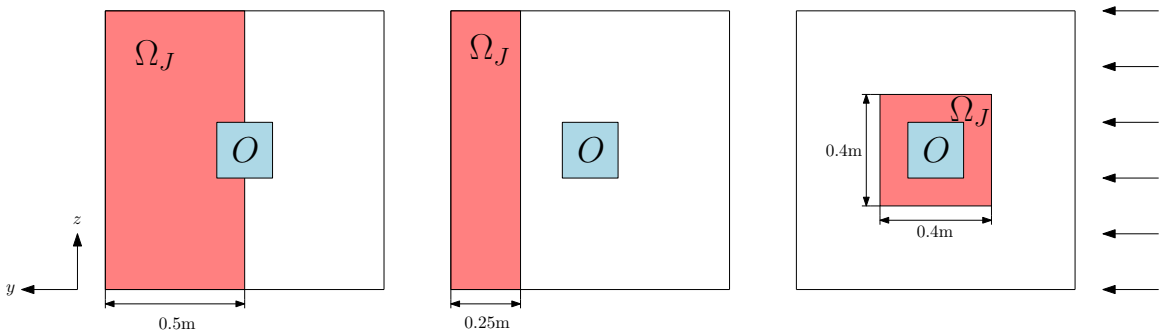


Figure 7.6: Classification measures for the cube identification case. The porosity threshold indicates the value from which the lattice-porosity is considered solid, i.e. $d_h = 0$ for $d_h <$ threshold.

threshold of $d_h = 0.9$ with 97.53%. Recall is highest for a high threshold with 99.87% at $d_h = 0.99$ in contrast to 99.09% at $d_h = 0.1$. For precision the opposite applies, with the highest value of 97.80% at $d_h = 0.1$ in contrast to 97.52% for $d_h = 0.99$. Although the results are very similar for the different thresholds, they show an expected trend. The higher the threshold is, the more of the object is found, and the lower the threshold is, the more certain the method is that a given node is solid. For the threshold $d_h = 0.9$, which corresponds to the highest Jaccard index, the method was able to correctly identify 9241 out of 9261 solid nodes, i.e. a recall of 99.78%. Also, 214 nodes which where identified as solid were actual fluid, which corresponds to 0.02% with respect to the total number of nodes. Note that the accuracy is expected to be very high because fluid nodes make up 99.98% of all nodes.

7.3.1 Partial data objectives

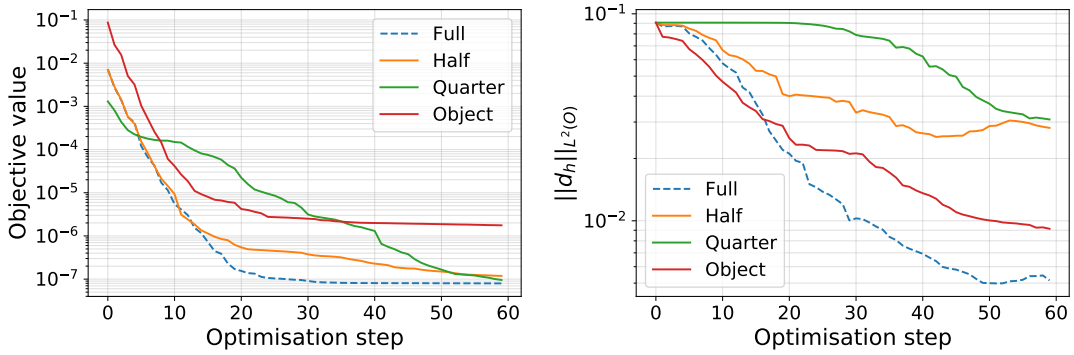
It is often the case with MRI that only partial data is available. To analyse the method in this respect, the objective domain $\Omega_J \subset \Omega$, i.e. the domain where the objective functional is being evaluated, is gradually reduced. The domains are selected to analyse the importance of the total available data compared to the localisation of the data. Three cases are considered here, with 50 percent of the total information being available (Half), 25 percent (Quarter), and 6.4 percent (Object), see Figure 7.7. In the Half and Quarter cases, the objective domain is located behind the object in flow direction. For the Object case, the objective domain is reduced to the design domain, which includes the object.



(a) Half (b) Quarter (c) Object

Figure 7.7: Test cases for partial data objectives. The information for the objective functional is reduced to a) 50 percent (Half), b) 25 percent (Quarter), and c) 6.4 percent (Object). Thereby the Half and Quarter domains are behind the object in flow direction, and the Object domain is around the to be identified object and corresponds to the design domain.

In [2] also different objective domains can be found.

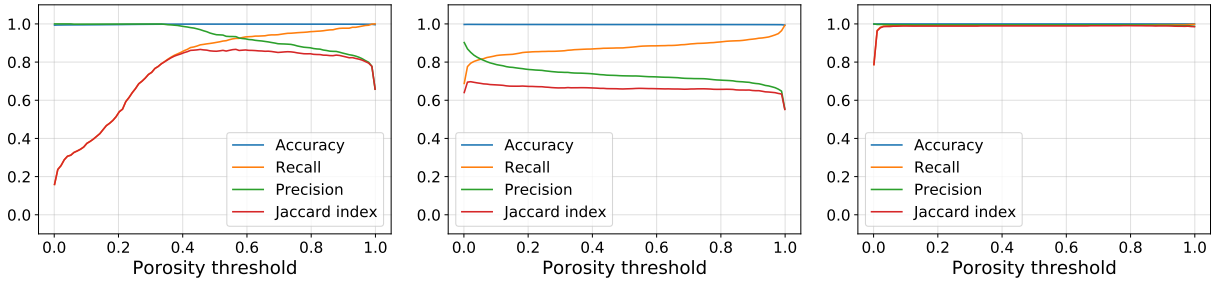


(a) Objective value (b) Porosity norm

Figure 7.8: Left, the objective value and right, the porosity norm in the object domain, for all partial information cases. The results of the full information case is shown as dashed line for comparison.

The comparison of objective value and porosity norm for the different cases are shown in Figure 7.8. The reduction of the objective value for the Half and Object cases follows a very similar trend to the Full case. In the Quarter case, the objective value falls much slower at the beginning, but reaches the lowest value among the reduced cases, but it also starts with the lowest value. The Object case starts and ends with the highest value. This difference in the objective value for the Object and Quarter case is due to the fact that in the Quarter case only the flow field outside the Object is available for the calculation where no major changes are expected. For the Object case, however, exactly that area is covered where the topology changes. The consideration of the objective value is therefore a useful indicator for the quality of the optimisation algorithm, but not for the object identification.

The porosity norm allows for an easier comparison because the changes are tracked in the same domain and thus also start with the same values. Here, the Object case has the lowest value among the partial cases, but is still above the Full case. Although the porosity norm decreases faster in the Half case than in the Quarter case, the value increases again after step 43 and both cases end at a similar value after 60 steps.



(a) Half (b) Quarter (c) Object

Figure 7.9: Classification measures for the partial information cases.

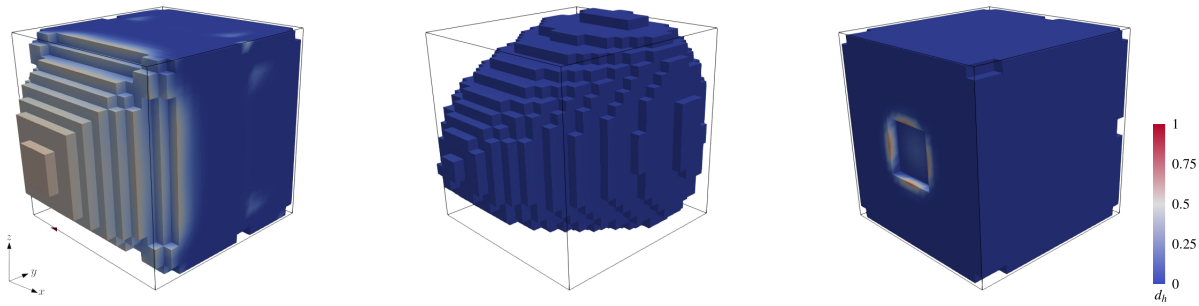
The classification measures for the different partial cases are shown in Figure 7.9. The Object case has the highest measures over all thresholds. For the Half and Quarter case precision rises from high to low thresholds and recall decreases. This is expected and was also found for the Full case above. For the Quarter case the Jaccard index stays around the same value, where for the Half case it drops for low thresholds.

∅	*Half*	*Quarter*	*Object*	*Full*
Accuracy	0.9978	0.9966	**0.9999**	0.9998
Recall	0.783	0.8758	0.9918	**0.9952**
Precision	0.9355	0.7348	**0.9948**	0.9776
Jaccard index	0.7237	0.6623	**0.9868**	0.9730

Table 7.3: Mean classification measures for all information cases over 100 thresholds from $d_h = 0.001$ to $d_h = 0.999$. The highest measures (bold) are found for the Object and the Full information cases, and the lowest for the Quarter information case.

Considering the mean values for the classification measures (cf. Table 7.3), the Object case is above 98% for all measures, and even higher than the Full case, only with the exception of recall. The Half and Quarter cases have a similar, high mean Jaccard index of 72.37% and 66.23%, respectively. The Quarter case has the lowest precision with 73.48%, where the Half case has the lowest recall with 78.83%. For the Half case the highest Jaccard index was found for $d_h = 0.57$ with 86.67%, for the Quarter case 69.71% for $d_h = 0.02$ and for the Object case 99.08% for $d_h = 0.78$.

In Figure 7.10 the visual representation of the different cases are shown for the last optimisation step and the highest Jaccard index, where the true object is marked by the black outline. In the Half case the cubical form of the object is found very well where



(a) Half (b) Quarter (c) Object

Figure 7.10: Visualisation of the identification results for the partial information cases with threshold of highest Jaccard index, as shown in Figure 7.9, with a) 86.67% for $d_h \leq 0.57$, b) 69.71% for $d_h \leq 0.02$, and c) 99.08% for $d_h \leq 0.78$.

the information is available, but in the front, where no velocity information is given, the geometry is not accurately identified. The same can be found for the Quarter case, where the method does not identify the correct geometry of the object, but nonetheless classifies nodes within the correct domain as solid, although no information in that area was given. The Object case identifies the true shape of the object very well, as was shown by the classification measures.

This results indicate that the method only finds the correct shape of the object where information is available, which was expected. But for the cases considered, the method is also able to detect the approximate position of an object to match the given flow field, even if information for the object itself is missing. Next, the same analysis is applied to a more complex object.

7.4 Complex object

The previous section has dealt with a simple, symmetric object. To further investigate the findings done there, a more complex, asymmetric object is now considered. This object is the Stanford bunny [143], which is an $3D$ scan of a ceramic rabbit figurine done by the Stanford University Computer Graphics Laboratory.

The Stanford bunny is placed in the middle of the same domain Ω as used in the previous section, but with a larger design domain, consisting of a cube with side length of 0.6 m, see Figure 7.11. The boundary conditions are the same as above.

The objective value, as shown in Figure 7.12, starts at 5.12×10^{-2} and is reduced by five-fold after five steps and reaches 4.68×10^{-6} after sixty steps. This relates to a relative error for the velocity field, measured in L^2 norm, which starts at 32.00% and ends with

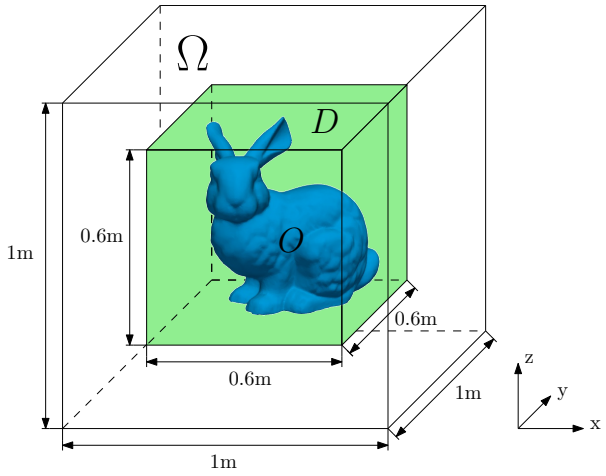


Figure 7.11: Setup for the Stanford bunny identification test case. The object to be identified is a $3D$ model of a ceramic rabbit [143], $O \subset \Omega$ placed inside the middle of the domain Ω. The design domain D, where the porosity will be changed, is a cube with side length of 0.6 m.

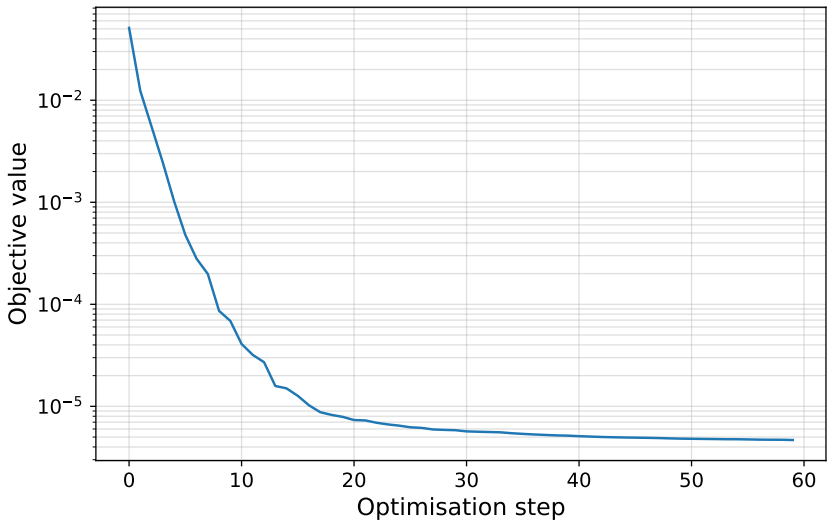


Figure 7.12: Plot of the objective value, $\frac{1}{2}||\boldsymbol{u}_f - \boldsymbol{u}^*||^2_{L^2(\Omega)}$, over the optimisation steps.

0.30% and thus a huge reduction of the error and a clear matching of velocity fields. The porosity norm inside the true object, $||d_h||_{L^2(O)}$, starts at 1.95×10^{-1} and is reduced to 6.99×10^{-2} (cf. Figure 7.13), and is a clear reduction.

Figure 7.14 shows the visualisation of the object identification, and after only 5 steps the outline of the Stanford bunny can be seen, with a clear representation after 50 steps. Again, like for the Cube identification case, a slice of the inside in the y–z plane is shown in Figure 7.15. The inside of the domain is getting low porosity values, but in contrast to the Cube case, there a still high porosity values behind the object in flow direction.

The classification measures (cf. Figure 7.16) show this trend, as high values are given for high porosity thresholds, which decrease with lower thresholds. This is expected, as

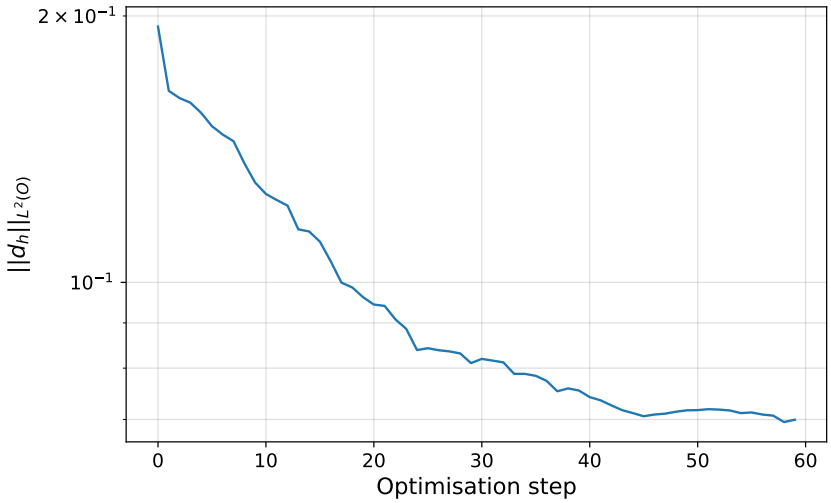
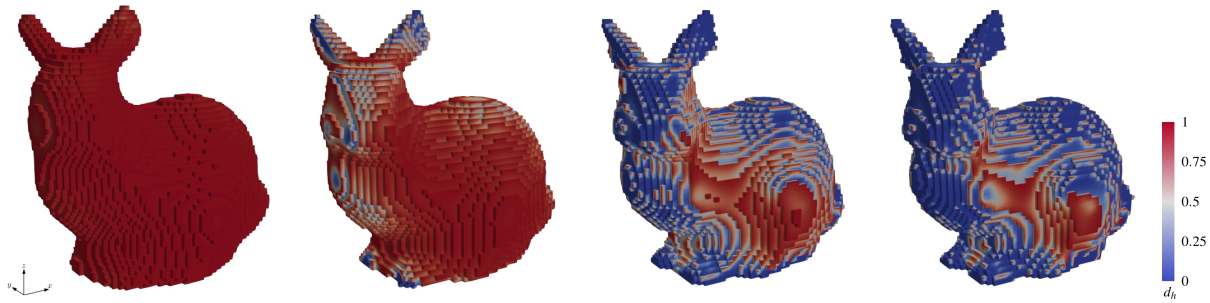


Figure 7.13: Plot of the porosity norm inside the object domain, $||d_h||_{L^2(O)}$, over the optimisation steps. The true object has a norm of $||d_h^*||_{L^2(O)} = 0$.



(a) 5 steps (b) 10 steps (c) 25 steps (d) 50 steps

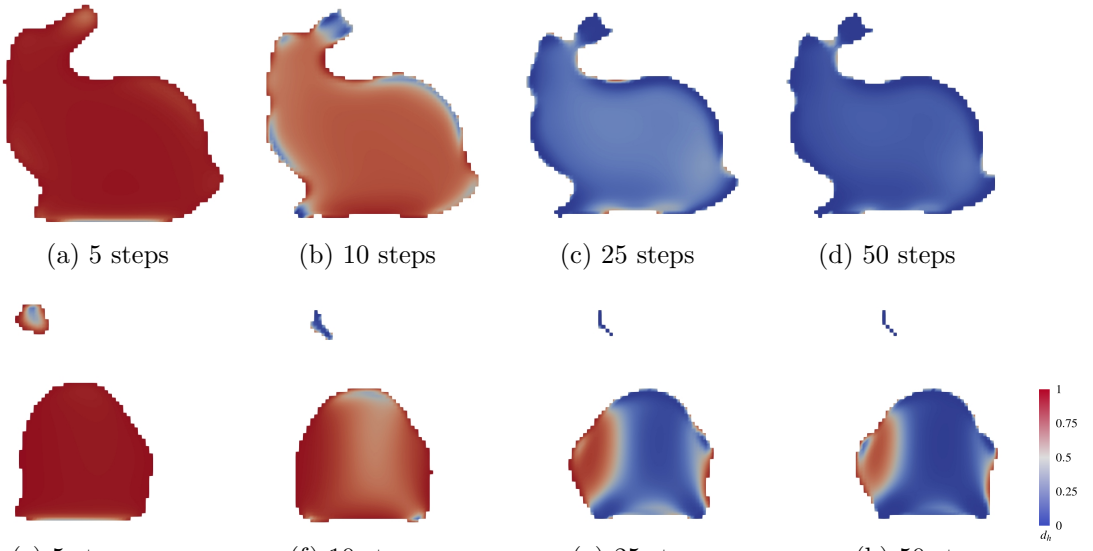
Figure 7.14: Visualisation of the Stanford bunny identification after 5, 10, 25 and 50 optimisation steps. The colour represents lattice-porosities for $d_h \leq 0.99$.

low thresholds would consider the area behind the object in flow direction as fluid nodes. Comparing the different classification measures for selected porosity thresholds, given in

Threshold	*0.1*	*0.5*	*0.9*	*0.99*	∅
Accuracy	0.9798	0.9963	**0.9993**	0.9987	0.993
Recall	0.4243	0.9003	0.9948	**0.9977**	0.8087
Precision	0.9888	**0.9923**	0.984	0.9651	0.9883
Jaccard index	0.4222	0.8941	**0.9790**	0.9629	0.8002

Table 7.4: Classification measures for the object identification with different thresholds of lattice-porosities for which the resulting object is considered to be solid. And the mean of 100 threshold values from $d_h = 0.001$ to $d_h = 0.999$.

Table 7.4, a similar trend to the Cube case can be seen. The highest Jaccard index is given for $d_h = 0.9$ with 97.90% showing a clear identification of the Stanford bunny. Here, 42455 of 42675 are correctly identified as solid nodes and 689 nodes are falsely classified as solid nodes, which is 0.06% of all nodes. Higher recall is again present for higher thresholds, where higher precision is found for lower thresholds. The low recall and Jaccard index for



(a) 5 steps (b) 10 steps (c) 25 steps (d) 50 steps

(e) 5 steps (f) 10 steps (g) 25 steps (h) 50 steps

Figure 7.15: Results of the Stanford bunny identification inside the object, shown in a slice of the cube in the y–z plane for a) - d) and in the x–z plane for e) - h) after 5, 10, 25 and 50 optimisation steps. The colour represents lattice-porosities for $d_h \leq 0.99$.

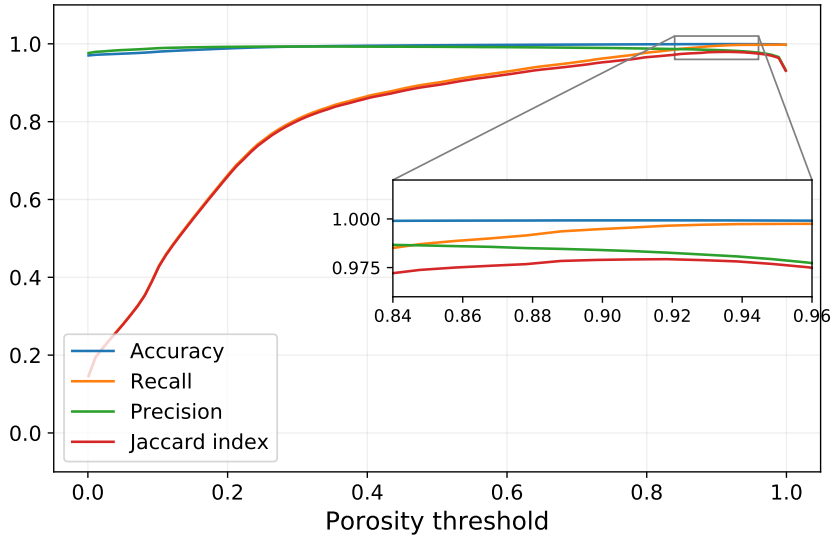
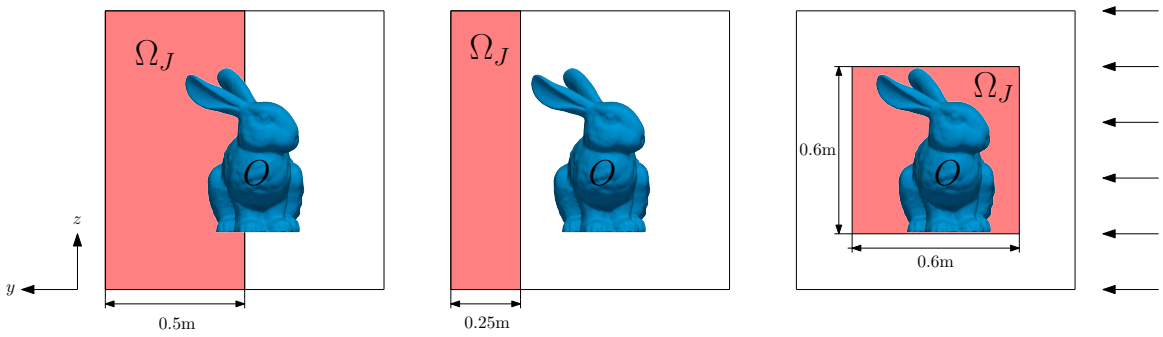


Figure 7.16: Classification measures for the Stanford bunny case. The porosity threshold indicates the value from which the lattice-porosity is considered solid, i.e. $d_h = 0$ for $d_h <$ threshold.

$d_h = 0.1$ are due to the high porosity values in the slipstream of the object.

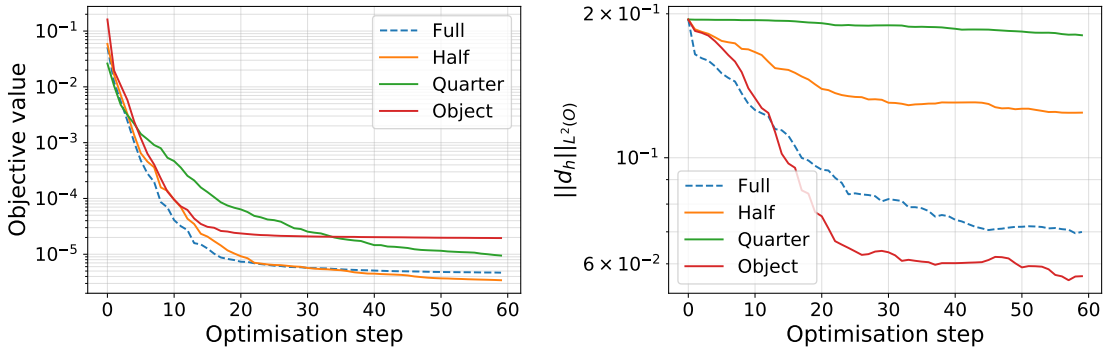
7.4.1 Partial data objectives

Again the object identification is tested for partially available data. The Half case, where the objective domain Ω_J is reduced to 50 percent, the Quarter case, where it is reduced to 25 percent and the Object case with 21.6 percent of the total domain, shown in Figure 7.17.



(a) Half (b) Quarter (c) Object

Figure 7.17: Test cases for the Stanford bunny partial data objectives. The information for the objective functional is reduced to a) 50 percent (Half), b) 25 percent (Quarter), and c) 21.6 percent (Object). Thereby the Half and Quarter domains are behind the object in flow direction, and the Object domain is around the to be identified object and corresponds to the design domain.

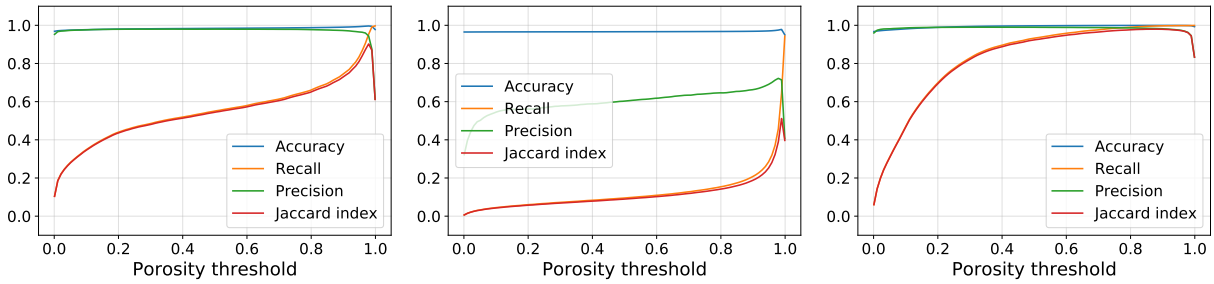


(a) Objective value (b) Porosity norm

Figure 7.18: Left, the objective value and right, the porosity norm in the object domain, for all partial information cases. The results of the full information case is shown as dashed line for comparison.

The results for the objective function and the porosity norm are very similar to the Cube identification case. The Objective value is reduced by the same amount for every case, showing that the optimisation method works very well. For the porosity norm inside the true object domain, the lowest porosity-values are given for the Object case, followed by the Full, Half and Quarter case. In contrast to the Cube identification case where Half and Quarter case had about the same porosity values in the last optimisation step, here the Half case has clearly lower porosity values than the Quarter case.

Figure 7.19 shows the classification measures based on the thresholds for the partial objective cases. A similar trend is seen for the three cases, where recall and Jaccard index decrease for lower porosity thresholds. Precision stays roughly constant but at different values, but is significantly lower for the Quarter case. The Object case has the highest



(a) Half (b) Quarter (c) Object

Figure 7.19: Classification measures for the partial information cases.

measures of all cases, as expected.

∅	*Half*	*Quarter*	*Object*	*Full*
Accuracy	0.9837	0.9667	**0.9936**	0.993
Recall	0.5524	0.1249	**0.8312**	0.8087
Precision	0.972	0.5974	0.9844	**0.9883**
Jaccard index	0.5396	0.1098	**0.8798**	0.8002

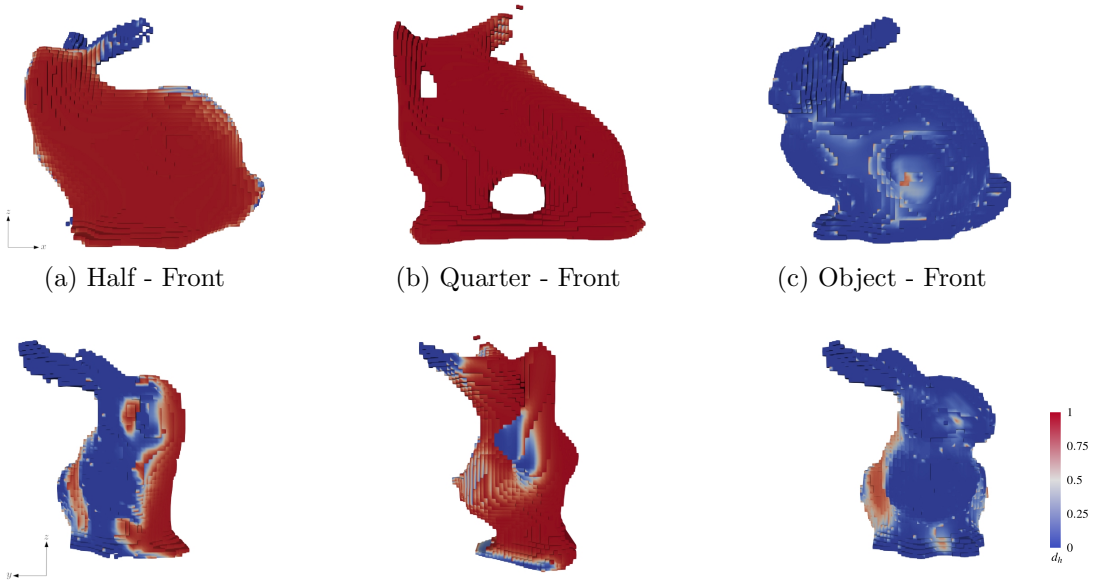
Table 7.5: Mean classification measures for all information cases over 100 thresholds from $d_h = 0.001$ to $d_h = 0.999$. The highest measures (bold) are found for the Object and the Full information cases, and the lowest for the Quarter information case.

Considering the mean of the measures for the partial and full cases (cf. Table 7.5), the object case has the highest values with the exception of precision, which is highest for the Full case. The Quarter case has the lowest measures, with a mean Jaccard index of only 10.98%. The highest Jaccard index for the Half case was 90.24% for $d_h = 0.98$, 53.06% for $d_h = 0.99$ for the Quarter case, and 97.97% for $d_h = 0.88$ for the Object case .

The low Jaccard index of the Quarter case can be seen in the visualisation, see Figure 7.20. Here the shape of the Stanford bunny can not be identified, but the location is found reasonable well. The Half case shows the same behaviour for the Stanford bunny as for the Cube case. Where information is available the shape is found very well, and where no information is available, i.e. in the front, the method identifies the location but nearly all details are lost. The Object case, which has the lowest total information available, but covers the true object, performs best. Here, the shape and localisation of the true object are found very precisely.

7.5 Conclusion

The aim of this chapter was to identify objects in a flow using only given flow data. A simple, symmetrical and a more complex, asymmetrical object were investigated for



(a) Half - Front (b) Quarter - Front (c) Object - Front

(d) Half - Side (e) Quarter - Side (f) Object - Side

Figure 7.20: Visualisation of the identification results for the partial information cases with front view on top and side view on the bottom. The threshold is chosen for the highest Jaccard index, as shown in Figure 7.19 with a) 90.24% for $d_h \leq 0.98$, b) 53.06% for $d_h \leq 0.99$, and c) 97.97% for $d_h \leq 0.88$.

different given data. For the analysis the objective functional, the L^2 norm of porosity, and the binary classification were used. The objective functional showed the quality of the optimisation algorithm very well, since its value decreased very fast and almost identically in all observable cases. However, it was not a good measure for the quality of the object identification. More suitable there was the porosity norm. Although a strong decrease of the value was a good indicator for the identification quality, a qualitative statement about it could only be made by employing binary classification. Using different measures, here accuracy, recall, precision and Jaccard index, the outcome of the object identification could be quantified. Accuracy was very high in all cases, as the objects occupied only a small percentage of the total domain. Generally, recall, the measure of correctly identified solid of all nodes, increased for high thresholds. Precision, the measure of correctly identified solid nodes, increased for low thresholds. This suggests that the porosity value is a measure of the certainty of the method to identify the solid nodes, i.e. the lower the porosity value, the less of the object is found but the more certain the method is that the object occupies this area. The Jaccard index is the measure of intersection over union of the identified and true object. For the full information cases the simple object had a Jaccard index of 97.53%, and the complex object 97.93%. For the partial cases it was found that the simple object has higher Jaccard indexes than the complex object. Also, details of the objects are lost, were no information is available, which was clearly seen in the complex object with 10.98% mean Jaccard index in contrast to 66.23% for the simple object.

Chapter 8

Noise reduction of flow MRI measurements

Noise reduction in MRI is a very active field of research and has produced a vast amount of different techniques to tackle that challenge. Classical methods are, for example, filtering methods like spatial and temporal [151], anisotropic diffusion [152], or nonlocal means filtering [153]. Other methods are transformation methods like the wavelet [154] and the curvelet transform [155], or statistical methods like maximum likelihood estimation [156] to only name a few. Even deep neural networks have been used to denoise images [157]. For a very good overview of advantages and limitations of classical methods, the work of Mohan et al. [110] is recommended.

In flow MRI many methods try to consider the underlying flow dynamics, for example by projecting the MRI data onto divergence-free wavelets [158], or by regularising curl and divergence-free solutions [159]. In order to consider mass and momentum conservation, i.e. the Navier–Stokes equations, proper orthogonal decomposition projects the solutions on a noise-free basis by sampling computational fluid dynamic simulations [160, 161]. Another method is the use of Tikhonov regularisation to apply Navier-Stokes equations to the measurement results [162]. Direct coupling of Navier-Stokes equations and MRI via data assimilation is primarily used for medical applications [6–9]. In these methods a boundary control problem is applied in which the geometry is known a priori. In other applications, the MRI data is used to obtain velocity profiles or to reconstruct geometries in patient data to set up CFD simulations [163–166].

However, in applications where the geometry of the imaged objects is complex, for example in filter applications, these methods can either be applied with difficulty or only with the help of computer tomography (CT), as either the flow is too noisy to use it as inflow

parameter, or the imaged objects are too complex to replicate it in the simulation.

This chapter has been published in [3].

8.1 Data generation

In order to test the CFD-MRI method, synthetically generated data is used, where noise is added to a simulation result. This way, the noise-free data is known and an extensive analysis can be performed. To not lose generality, the data is matched with MRI data, which was acquired by Pro^{2}NMR at KIT.

8.1.1 MRI data

The velocity noise $\sigma_{velocity}$ can be estimated from the two images, which are usually measured to calculate the velocity image. First, an area where no signal is expected has to be chosen to get the typical statistical noise in an image. In this noise area the mean value $\mathrm{SNR}_{\mathrm{mean}}$ has to be calculated, which represents the noise in the intensity image. With $\mathrm{SNR}_{\mathrm{mean}}$ and v_{enc} the velocity noise $\sigma_{\mathrm{velocity}}$ can be estimated by (cf. Section 3.1):

$$\mathrm{SNR}_{\mathrm{mean}} = \frac{I}{n_{\mathrm{mean}}}, \qquad \sigma_{\mathrm{velocity}} = \frac{\sqrt{2} v_{\mathrm{enc}}}{\pi\, \mathrm{SNR}_{\mathrm{mean}}},$$

with pixel intensity I and mean noise n_{mean} [107, 167, 168]. Another way to calculate the velocity noise is to pick a line in the velocity image, where the velocity is constant. Along this line, the standard deviation v_{std} is calculated and used for the velocity noise estimation. In these experiments, the outlet of the measurement object was chosen, where an almost constant volume flow is expected. In Section 8.1.3 these velocity noise values (cf. Table 8.2) are used to generate synthetic data.

	High Noise	**Low Noise**
$x-$Resolution	$62.5\,\mu$m	$62.5\,\mu$m
$y-$Resolution	$62.5\,\mu$m	$62.5\,\mu$m
Slice Thickness	1 mm	1 mm
Number of Acquisitions	2	16
Scan Time	0 h 13 min	3 h 24 min
v_{std}	4.197×10^{-3}m/s	0.729×10^{-3}m/s

Table 8.1: Properties of the MRI data, for the setup see Figures 8.1 and 8.2

For the properties of the MRI data see Table 8.1.

8.1.2 Simulation data

In order to generate the synthetic data a three-dimensional simulation was carried out which matches the physical setup of the MRI measurement. The geometry of the domain is therefore a pipe with 5.7 mm diameter and length of 17 mm, with cylindrical object at position 7 mm in y direction, 2.3 mm in x direction, and with a diameter of 1 mm, see Figure 8.1. The fluid is isopropanol with kinematic viscosity of $2.798 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$ and density of $786\,\mathrm{kg}/\mathrm{m}^3$. The simulation has a Poiseuille inflow with $0.75 \times 10^{-2}\,\mathrm{m/s}$ mean velocity, free outflow and no-slip condition at the wall. Here, a start value for the permeability in the domain (cf. [1]) of $K = 1.0 \times 10^{-7}\,\mathrm{m}^2$ was chosen, which showed a sensitive behaviour for the simulation of porous media [70], see also Sections 2.3.3 and 5.6. If the initial value is chosen too low or too high, the sensitivity is reduced and the gradients required for the optimisation are too small to achieve real changes (cf. [2]). Although a step condition, here the Wolfe conditions, can prevent too small step sizes and gradients for the optimisation method, a start value should be chosen so that the underlying porous medium simulation is valid [2]. The grid contains around 2 million lattice nodes with a spatial resolution of 62.5 μm, matching the MRI data. It is important to note that the experimental measurement was performed in the middle of the pipe. In general, however, the method also works for data with any position, size and dimension (cf. [1, 2]). Thus the simulation domain is independent on the available data, as seen in the following numerical experiments.

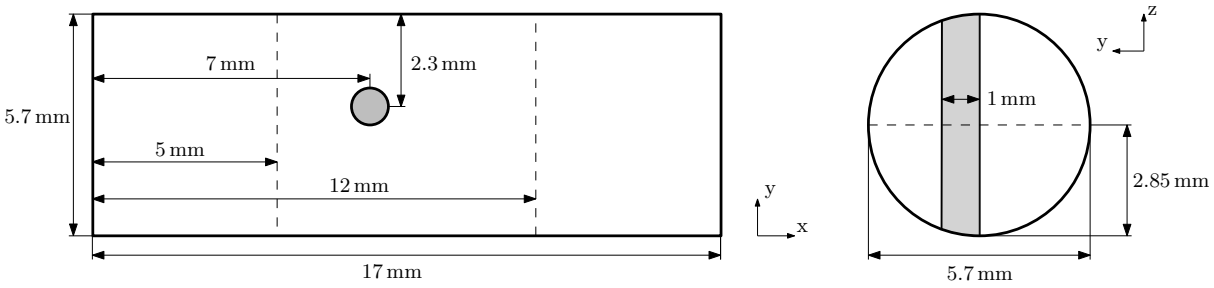


Figure 8.1: Setup for the simulation, matched to the MRI setup, with cylindrical object placed inside a pipe. The location of the MRI data is indicated by the dashed lines.

8.1.3 Comparison of MRI and synthetic data

In order to generate synthetic data, Gaussian noise (cf. Section 3.2) with mean zero and standard deviation corresponding to the velocity noise values of Table 8.1 is added to the simulation result, and compared to the MRI data. In Section 8.2, this synthetic data is then used in the objective function to apply the CFD-MRI method to it (cf. Algorithm 3).

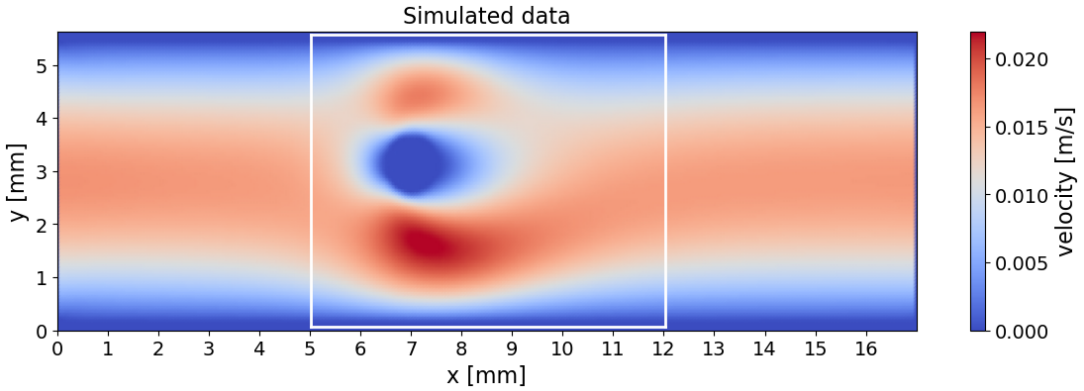


Figure 8.2: Slice of the simulation result in the x–y plane in the middle of the domain. The location of the MRI data is indicated by the white rectangle (cf. Figure 8.1)

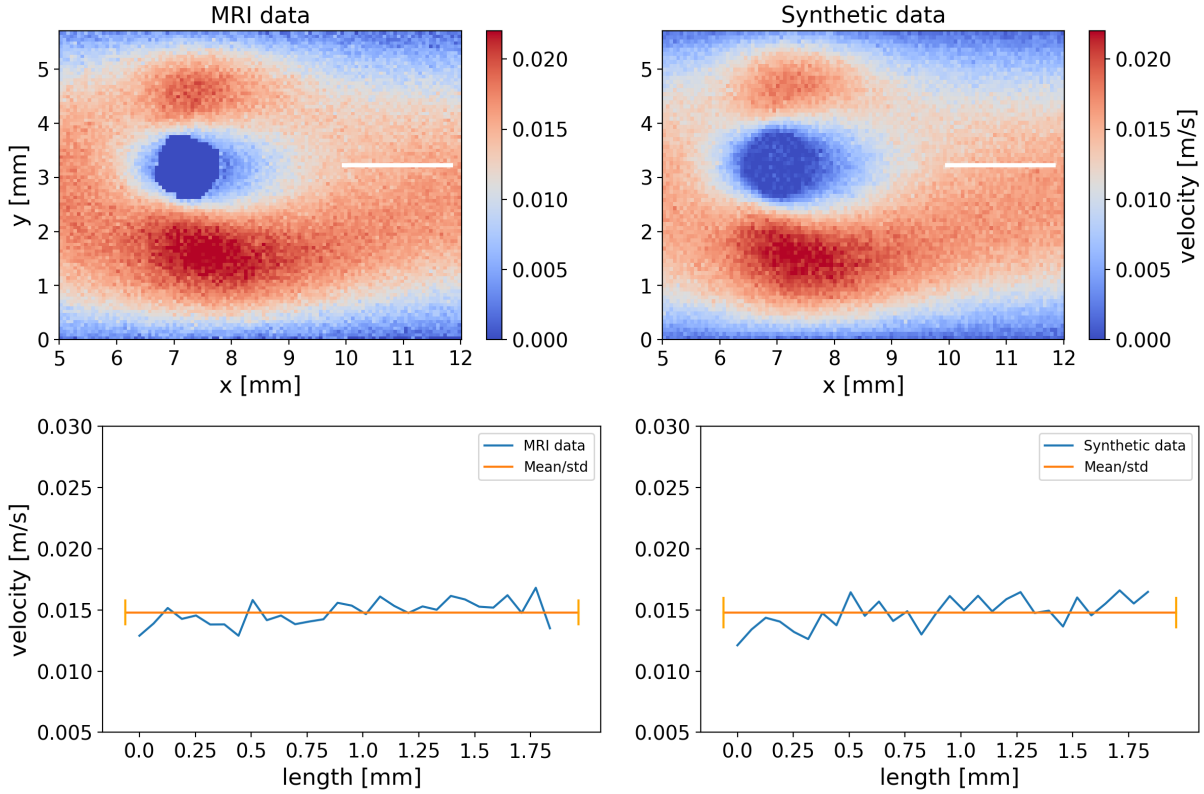


Figure 8.3: On top, visual comparison of MRI data and synthetic data, both with low noise. The magnitude velocities along the white line with mean and standard deviation, at the bottom.

In Figures 8.3 and 8.4 the comparison of MRI data and synthetic data for the given noise power is shown, where the white line represents the measurement of the velocity noise, as discussed in Section 8.1.1. The calculated mean and standard deviation for synthetic, MRI and simulation data is shown in Table 8.2. The noise power values match very well, with $1.0 \times 10^{-3}\,\mathrm{m/s}$ to $0.9 \times 10^{-3}\,\mathrm{m/s}$ for the low noise data and $5.4 \times 10^{-3}\,\mathrm{m/s}$ for the high noise data. The mean velocity ranges from $1.45 \times 10^{-2}\,\mathrm{m/s}$ to $1.51 \times 10^{-2}\,\mathrm{m/s}$ in all cases. Thereby, the difference from synthetic to MRI data for low and high noise data

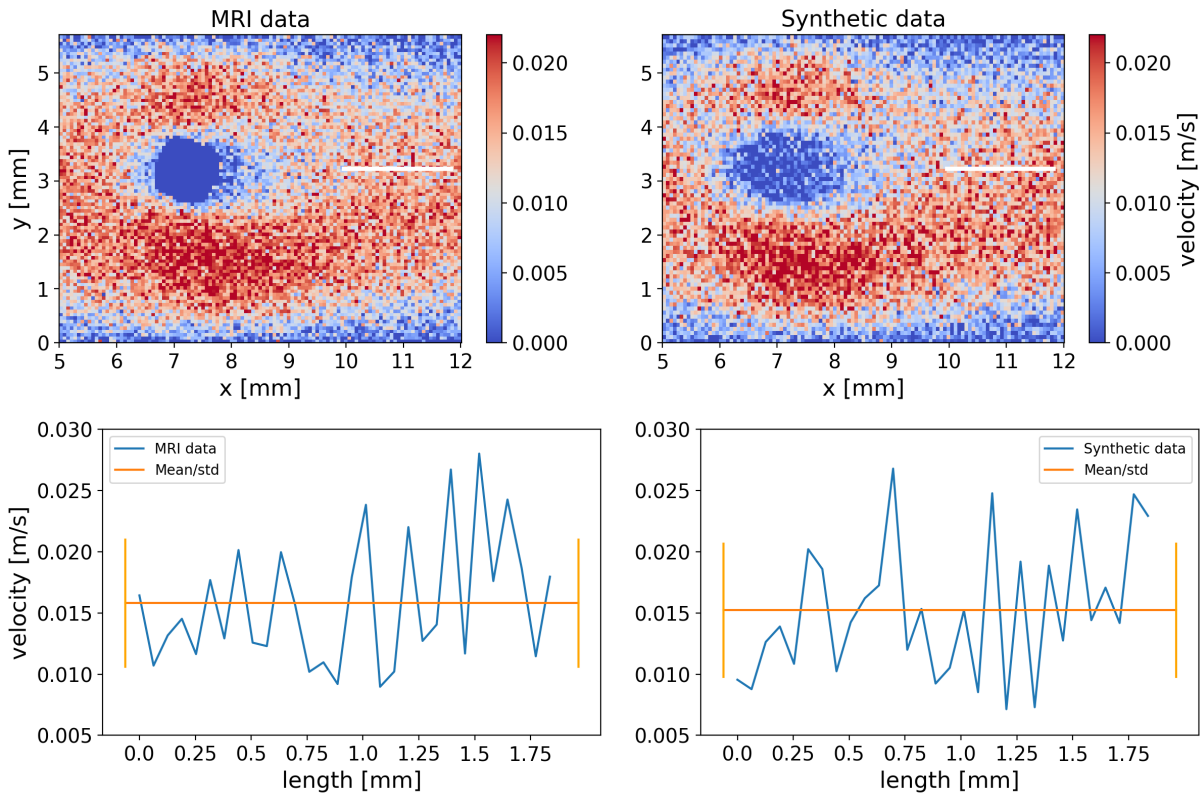


Figure 8.4: On top, visual comparison of MRI data and synthetic data, both with high noise. The magnitude velocities along the white line with mean and standard deviation, at the bottom.

is 0.04×10^{-2} m/s, which is very close to the difference between low and high noise MRI data of 0.02×10^{-2} m/s. This small difference in mean velocities from synthetic to MRI data may be due to the object in the flow, which is a perfect cylinder in the simulation, but not in the MRI data. The synthetic data also poses an additional challenge to the method, as there is noise in the area of the object, which is not the case with MRI data, especially seen in the high noise data (cf. Figure 8.4). Here, it was chosen not to remove the additional noise in order to see how the method handles this.

Noise Power	**Data**	**Mean**	**Std**
Low	MRI	0.0151	0.0010
	Synthetic	0.0147	0.0009
High	MRI	0.0149	0.0054
	Synthetic	0.0145	0.0054
None	*Simulation*	0.0147	0.0006

Table 8.2: Mean and standard deviation in m/s of MRI and synthetic data for low and high noise along the white line in Figures 8.3 and 8.4.

8.2 Analysis of noise reduction using CFD-MRI

The CFD-MRI method is applied to the synthetically generated data (cf. Section 8.1). Thereby, the data has low and high noise power with incremental reduction of available spatial and velocity information. The analysis is performed using visual representation, comparison of velocity plots and error norms. Thereby, the L_h^2 norm $||\cdot||_{L_h^2(\Omega_h)}$ is used, defined as

$$||\mathbf{u}||_{L_h^2(\Omega_h)} = \left(\sum_{u \in \mathbf{u}} |u|^2 \right)^{1/2}, \tag{8.1}$$

for velocity data $\mathbf{u}$ in domain Ω_h. The data corresponds to simulation result, synthetic data, MRI data, or a combination. The relative error between two velocity data $\mathbf{u}$ and $\mathbf{v}$ is defined as

$$\frac{||\mathbf{u} - \mathbf{v}||_{L_h^2(\Omega_h)}}{||\mathbf{u}||_{L_h^2(\Omega_h)}}, \tag{8.2}$$

where the subtraction is considered to be pointwise.

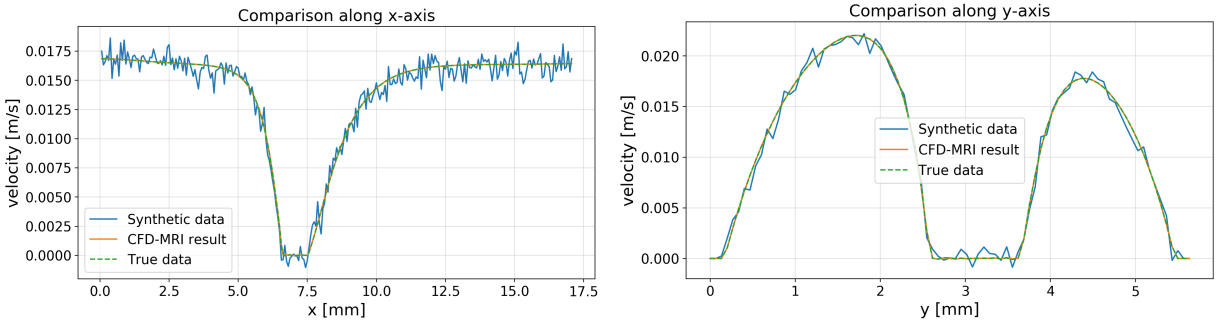
The optimal control problem was solved on the high-performance computer ForHLR II at KIT for three days on 200 cores with Deca-Core Intel Xeon E5-2566 v3, which led to about 30 optimisation steps for every test case. Note that the computation time does not change for different data, i.e. if low or high noise data is used, but only by the resolution of the simulation.

8.2.1 Low noise data

Synthetic data with low noise corresponding to a measurement time of 3 h and 24 min (cf. Section 8.1), is considered first. The data has an error of 13.44%, measured as relative error of synthetic data to simulation data in L_h^2 norm (Table 8.3). The CFD-MRI result, i.e. the synthetic data after being applied to CFD-MRI, reaches an error of only 0.17%, which is nearly eighty times less than the original data. The lowest error is in the main flow component of 0.12%. In Figure 8.5 velocity plots along the x and y axes in the middle of the domain are shown, with synthetic data, noise-free data and CFD-MRI result. The noise-free data and the CFD-MRI result thereby coincide almost exactly.

Error	Total	X	Y	Z
Synthetic data	13.44 %	56.75 %	07.86 %	97.26 %
CFD-MRI	00.17 %	00.93 %	**00.12 %**	03.38 %

Table 8.3: Relative error, measured in L_h^2 norm, of low noise synthetic data before and after being applied to CFD-MRI, to noise-free data. Shown is the total error and the error for each velocity component. The high errors in the x and z components in the synthetic data is due to the small velocity magnitudes compared to the added noise power.



(a) (b)

Figure 8.5: Plots for synthetic data, CFD-MRI result and noise-free data along x and y axes in the middle of the domain. The noise in the data is almost completely removed, with a 0.12% relative error between result and noise-free data.

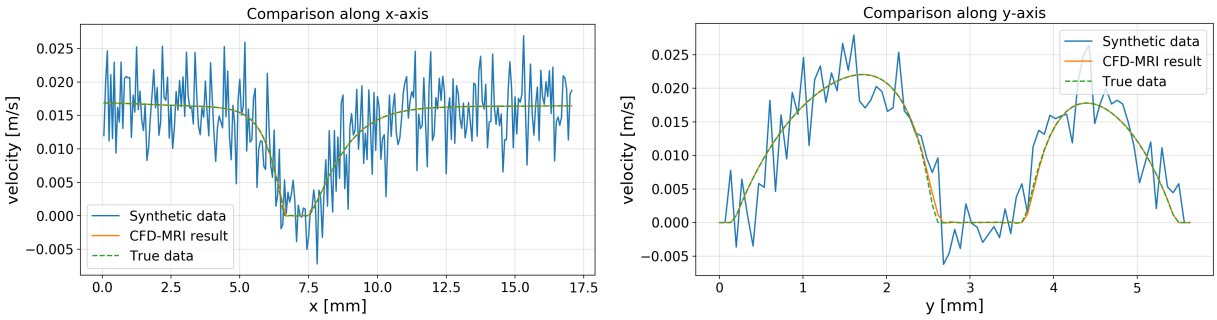
8.2.2 High noise data

To provoke the limits of CFD-MRI, the method was applied to the synthetic data with an unreasonably high noise figure, corresponding to a measurement time of only 13 min. Here, the difference from synthetic data to noise-free data is very high, up to 61.51 %

Error	Total	X	Y	Z
Synthetic data	61.51 %	96.94 %	41.28 %	99.91 %
CFD-MRI	00.68 %	03.52 %	**00.50 %**	12.86 %

Table 8.4: Relative error, measured in L_h^2 norm, of high noise synthetic data and its CFD-MRI result to noise-free data. Shown is the total error and the error for each velocity component. The high errors in the x and z components in the synthetic data is due to the small velocity magnitudes compared to the added noise power.

(cf. Table 8.4 and Figure 8.6). After applying the CFD-MRI method, the total error is only 0.68%, which is about ninety times less than the noisy input data (cf. Table 8.4). For the main flow component in y direction it is even less with 0.5 %. When comparing the results visually (cf. Figure 8.7), it can be clearly seen how the high noise in and around the object will be completely removed.



(a) c (b) c

Figure 8.6: Plots for synthetic data, CFD-MRI result and noise-free data along x and y axes in the middle of the domain. The extreme noise in the data is almost completely removed, with a 0.68% relative error between result and noise-free data.

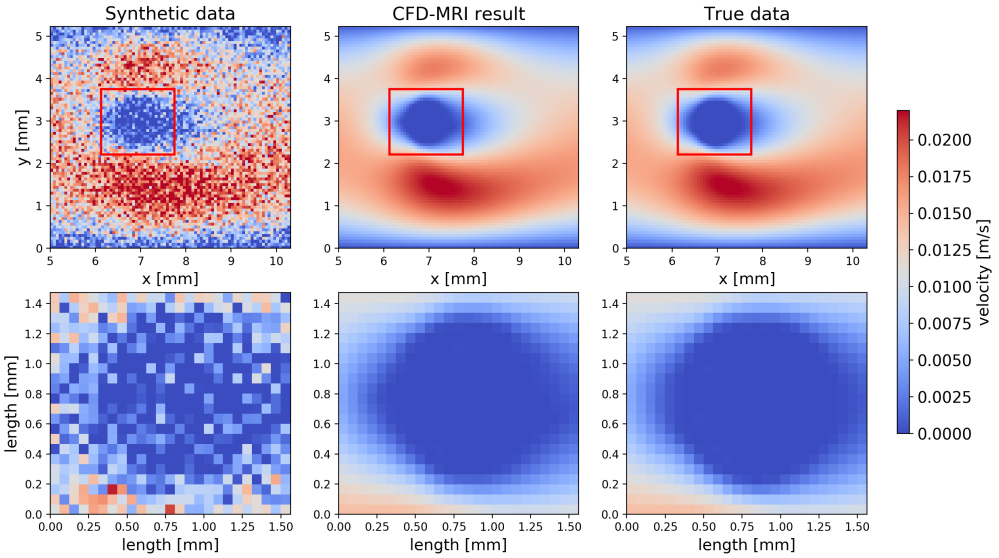


Figure 8.7: From left to right, visual representation of synthetic data, CFD-MRI result and noise-free data with a zoomed in image around the object below. Even with high noise in the object the difference of CFD-MRI result and noise-free data is very low with only 0.68% relative error.

8.2.3 Reduced information

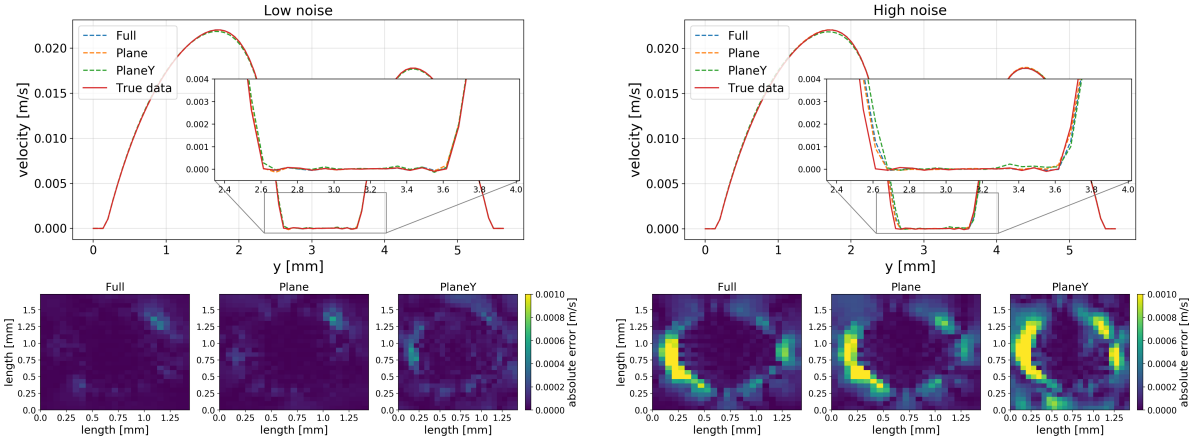
Here, the information available to the CFD-MRI method will be reduced gradually. First, full information, i.e. $3D$ spatially resolved with all three velocity components, here called *Full* is used. Next, the data is reduced to $2D$ spatially resolved data, corresponding to the MRI plane (cf. Figure 8.2), called *Plane*. The last reduction will match the MRI data, by having $2D$ spatially resolved data with only the y velocity component, i.e. the main component along the long axis of the pipe, called *Plane Y*. In all cases, the error is measured with data locating in the MRI plane with y component of the velocity.

As expected, in all test cases the error for the low noise data is lower than in the high noise data. Furthermore, and as well expected, the less information available, the higher the error. In the low noise data, the error is between 0.13% and 0.42%, whereas the high noise data does not have much larger errors, between 0.56% and 1.21%. In all cases the

Data	*Full*	*Plane*	*PlaneY*
Low noise	0.13%	0.16%	0.42%
High noise	0.56%	0.66%	1.21%

Table 8.5: Relative error, measured in L^2_h norm, for the different information cases with low and high noise data. The less information available and higher the noise, the greater the error, as expected. In all cases, the error is calculated in the MRI plane (cf. Figure 8.1) for the y velocity, i.e. the main component of the flow.

highest errors are around the object, which can be seen in Figure 8.8.



(a) Low noise data

(b) High noise data

Figure 8.8: Plot along the y axis, i.e. through the object along the diameter of the pipe, with zoomed in sections around the object on top. And the visual representation of the absolute error to noise-free data for the information test cases *Full*, *Plane* and *PlaneY* on the bottom.

When comparing the total error, i.e. for three dimensions and all velocity components, a similar result can be seen. For the high noise test the results are 3.55% if only the information in the plane is used as comparison data and 9.38% if only the information in the plane and additionally only the y-velocity are known. This is an enormous reduction compared to the 78.02% error of the noisy data. Very similar are the results for the low-noise data, where the error is 2.86% if only the data in the plane is given, and 7.36% for the data in the plane and the y-component. Note again that these errors are in the entire simulation domain and for all velocity components, not just where the information was given.

8.3 Application to real MRI data

In this section, the method is applied to the real data. By comparing the results for MRI data with different noise power, the performed analysis for synthetic data (cf. Section 8.2) is investigated.

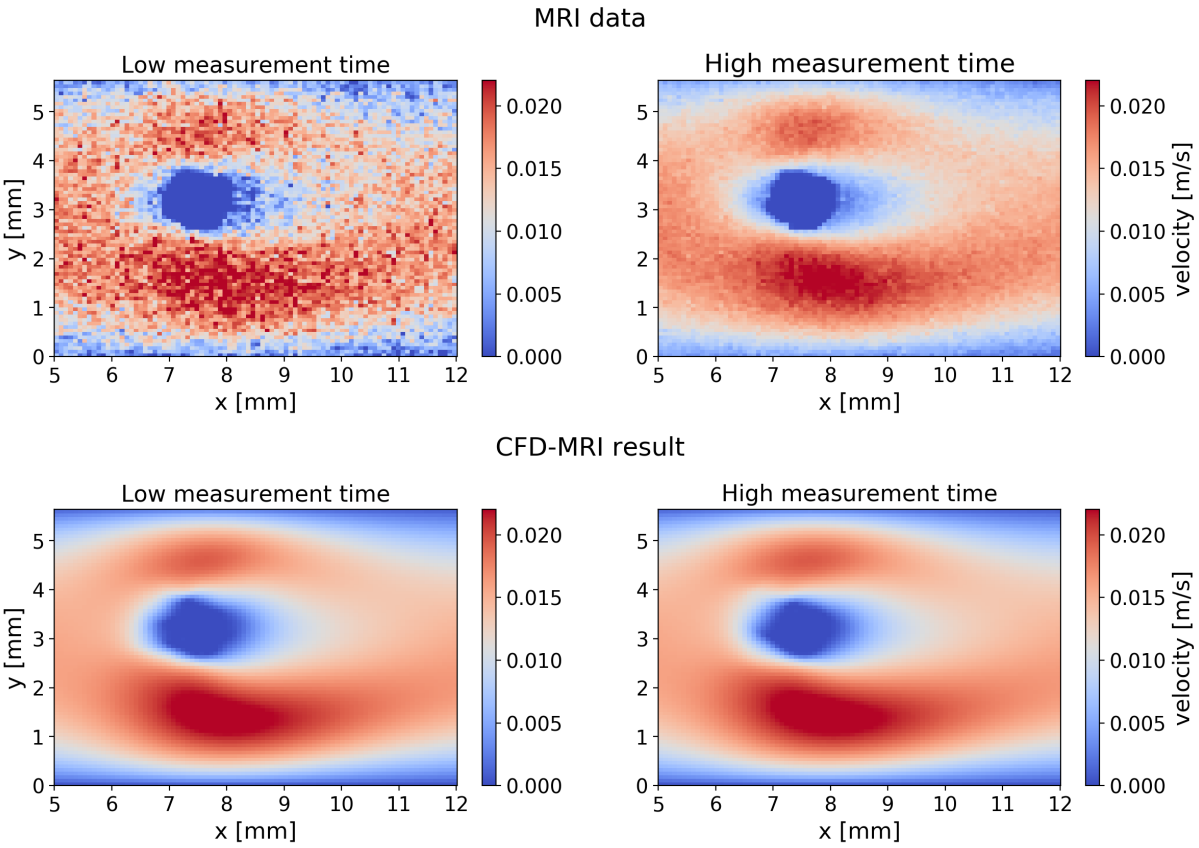


Figure 8.9: MRI images and CFD-MRI result for different acquisition times. The results of the CFD-MRI method differ by only 1.73%, even with six times difference in noise power of the input data.

The relative error between the two results of the CFD-MRI method is only 1.73%, even with approximately six times difference in noise power of the input data. This is a good indication that the results from Section 8.2 are applicable to real data. It should also be noted that the scan time for the MR images differs by more than three hours, here about 16 times (cf. Table 8.1). Thus, scan time can be dramatically reduced without having a significant impact on the result (cf. Figures 8.9 and 8.10).

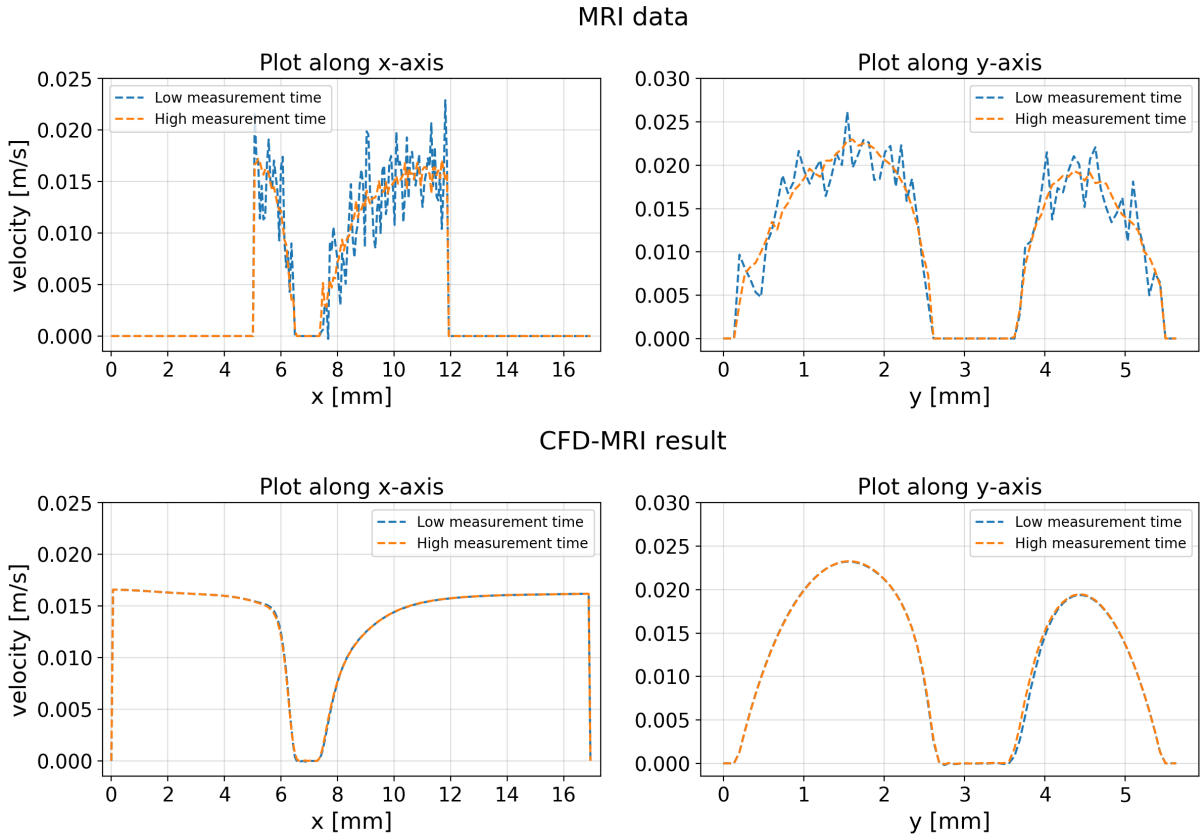


Figure 8.10: Plot over lines along x and y axes of the data for different acquisition times. At the top, the different MRI data, and on the bottom the corresponding CFD-MRI result. The two results differ by only 1.73% in the whole plane, even with six times difference in noise power of the MRI data. Note that the simulation can be performed on a larger domain, as shown in the plot along the x axis of the CFD-MRI result compared to the MRI data.

8.4 Conclusion

In this chapter the CFD-MRI method was analysed for its ability to reduce noise. In order to analyse the method, MRI data was used to generate synthetic data. With it, the noise-free, true data was available and analysis for different given spatial and velocity information as well as noise power was carried out. As a result, it was found that the method is not very dependent on intensity of the noise for the given data. With less noisy data the error was as low as 0.17%, which was eighty times less than the original noisy data. And even with limited information available and high noise in the data, the error to noise-free data did not exceed 1.5%. A comparison with real data indicates thereby the correctness of the analysis with artificial data. Here, the results of the method differed by only 1.73%, despite a six-fold difference in the noise power of the MRI data. Thus, one advantage of the combined method is that much faster MRI scans can be performed with less effort on increasing the SNR. Therefore a clean image can be derived by only having minutes of scan time, in contrast to hours. Another application is using highly spatially

resolved images, with a low SNR in the flow field. These images have a much clearer representation of the object or objects in the flow, and the noise can still be removed using CFD-MRI.

Chapter 9

CFD-MRI applied to flow through a porous structure

In this chapter the CFD-MRI method is applied to flow through a porous structure, i.e. an object which contains voids and is able to transmit fluids (cf. Section 2.1.3). In the previous chapters only simple objects or simple flows were considered. To investigate the method to be usable in real-world applications, like filtration in water treatment or pollution control, or blood flow in arteries, a complex object with resulting complex flow and added noise is used here for the analysis of object identification and noise reduction. The structure thereby is a reconstruction based on a flow MRI measurement done by Mojtaba Mirdrikvand, Department of Chemistry, University of Bremen.

9.1 Setup

The setup is a pipe flow with radius of $r = 5.2 \times 10^{-3}\,\mathrm{m}$ and length $L = 29.6 \times 10^{-3}\,\mathrm{m}$. The fluid is water with density of $\rho = 998\,\mathrm{kg/m^3}$ and kinematic viscosity of $\nu = 1.0 \times 10^{-6}\,\mathrm{m^2/s}$. The design and objective domain for the optimisation correspond to the location of the porous structure as shown in Figure 9.1. The boundaries are set to Poiseuille inflow with maximum velocity of $1.0 \times 10^{-3}\,\mathrm{m/s}$ at $z = 0\,\mathrm{m}$, free outflow, i.e. constant pressure, at $z = 29.6 \times 10^{-3}\,\mathrm{m}$, and bounce back to the remaining boundary. The domain is discretised with $\triangle x = 0.1 \times 10^{-3}\,\mathrm{m}$ resulting in more than three million lattice nodes total. The relaxation time is chosen to be $\tau = 0.56$. And for the optimisation a start value of $K = 1.0 \times 10^{-8}\,\mathrm{m^2}$ is used.

Since the adjoint problem is solved using the same framework as the forward problem, the moments can be computed in the same way. Here the performance of the method

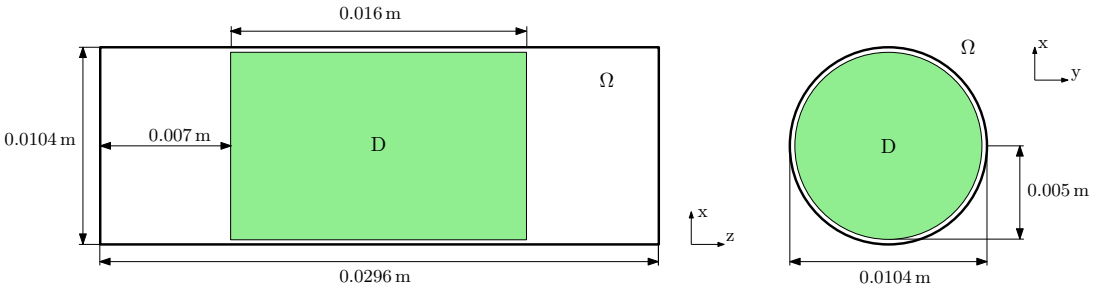


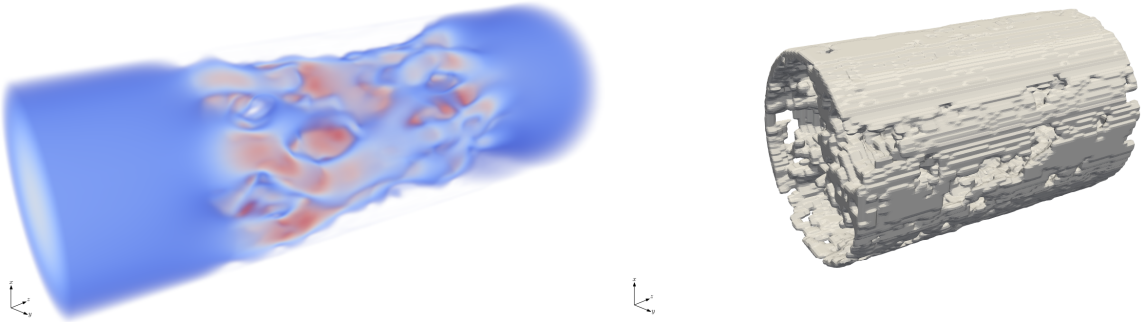
Figure 9.1: Simulation setup of the flow through the porous structure.

was increased by employing the steady state solution of the forward problem and the explicit time marching scheme of the LBM. This ensures that a convergence check of the adjoint moments, here the adjoint velocity, can be employed which massively reduces the computation time. Here convergence is claimed if the standard deviation of the maximum adjoint velocity is smaller than 1.0×10^{-4} times its average.

The problem was solved on 60 cores of Intel® Xeon® Silver 4114 with 2.20GHz.

9.2 Data preparation and analysis

For the data acquisition, first a simulation was performed with flow through a porous structure. The resulting flow field was then saved as synthetic data as done in Chapters 7 and 8. The visualisation of this data is shown in Figure 9.2. The flow field has a mean



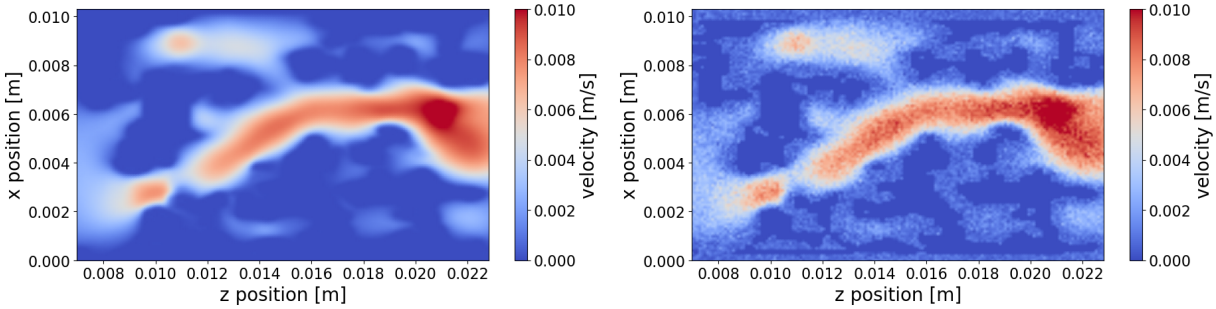
(a) Flow field (b) Porous structure

Figure 9.2: Visualisation of the true data of a) the flow field and b) the object.

velocity of 0.38×10^{-3} m/s with standard deviation of 1.09×10^{-3} m/s. The total number of nodes is $3,274,425$ with $2,683,993$ fluid and $590,432$ solid nodes. The design domain, which corresponds to the outline of the structure, consists of $1,261,715$ nodes, i.e. the object has a porosity of around 46.80%.

9.2.1 Synthetic noise

Here, synthetic noise is added to the simulation result. As seen in the previous chapters Gaussian noise can be used if the SNR is high enough (cf. Section 3.2), and also matches real data very accurately (cf. Chapter 8). Here, Gaussian noise with mean zero and standard deviation of 0.5×10^{-3} m/s is added to every velocity component, resulting in a relative L^2 error between true and noisy flow field of 38.55%. A visual comparison of the noisy and true flow field is shown in Figure 9.3.



(a) True data

(b) Noisy data

Figure 9.3: Slice in the x–z plane of a) the true data and b) the noisy data.

9.3 Results

The optimisation works very well with a fast decrease in the objective value, as shown in Figure 9.4. Starting from $J = 3.88 \times 10^{-1}$ the objective value drops fast for the first ten optimisation steps and converges to 0.4×10^{-1} after 50 steps. The same results were found without using the adjoint convergence check. Here, the objective values differed by less than 0.06%, but the performance increased more than five-fold with only 14 min compared to 75 min per optimisation step. It should also be noted that without using the curvature condition, i.e. only using the Armijo rule in the line search, it was found that the optimisation did not converge, and got stuck after only two optimisation steps at the starting value of 3.88×10^{-1}.

Since the noise-free flow and the geometry of the true object is known the results can be analysed with regard to noise reduction and object identification, which is done in the following.

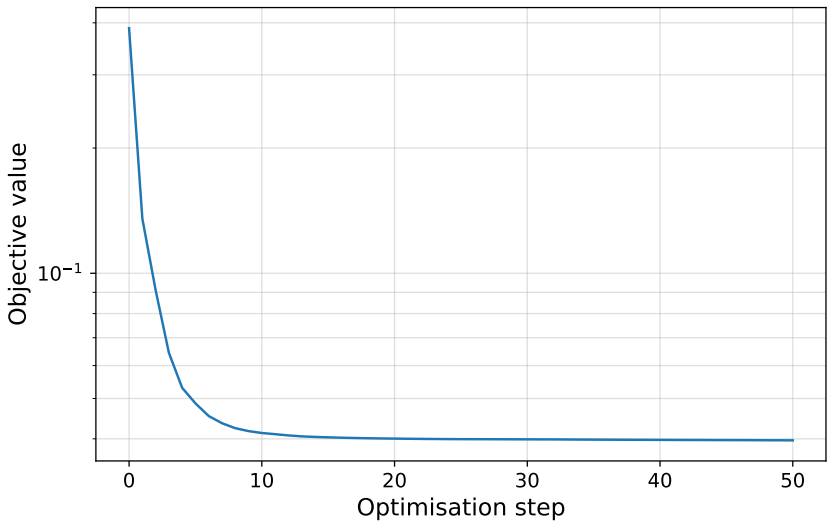


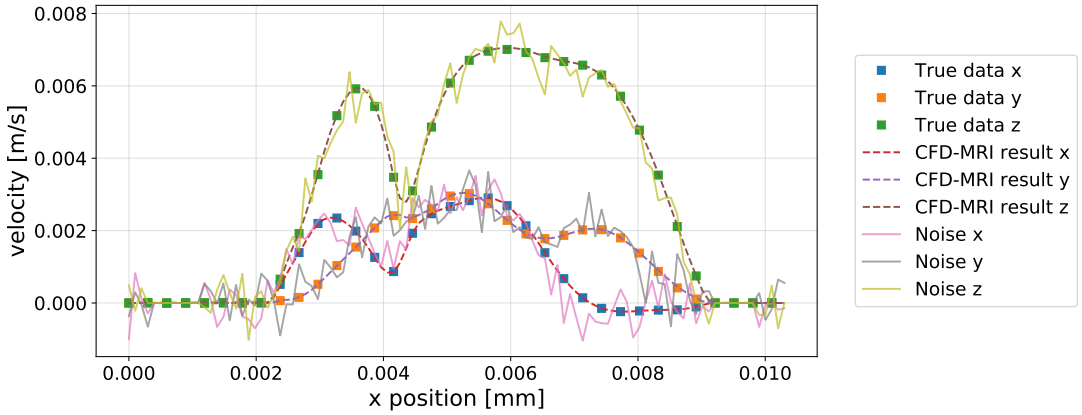
Figure 9.4: Value of the objective function, $\frac{1}{2}||\boldsymbol{u}_f - \boldsymbol{u}^*||^2_{L^2(\Omega_J)}$, of noisy data $\boldsymbol{u}^*$ to CFD-MRI result $\boldsymbol{u}_f$ over the optimisation steps.

9.3.1 Noise reduction

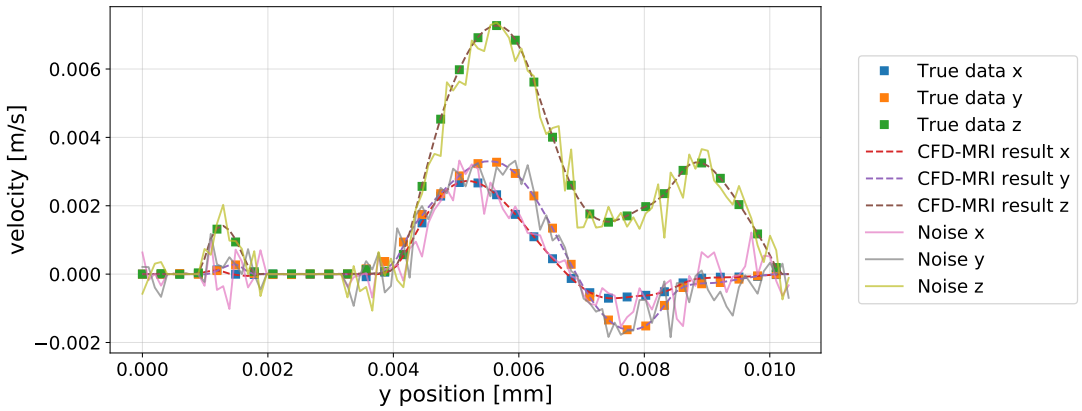
In Figure 9.5 the velocity profiles along a line through the middle of the domain in $x-$, $y-$ and $z-$direction are shown with all three velocity components for true data, noisy data, and CFD-MRI result. The results show a significant reduction in noise, while at the same time corresponding very well to the true data, in all directions and for every component of the velocity. The relative L^2 error from CFD-MRI result to the true data is 4.58%, which is a more than eight-fold reduction of the 38.55% error of the noisy data.

9.3.2 Object identification

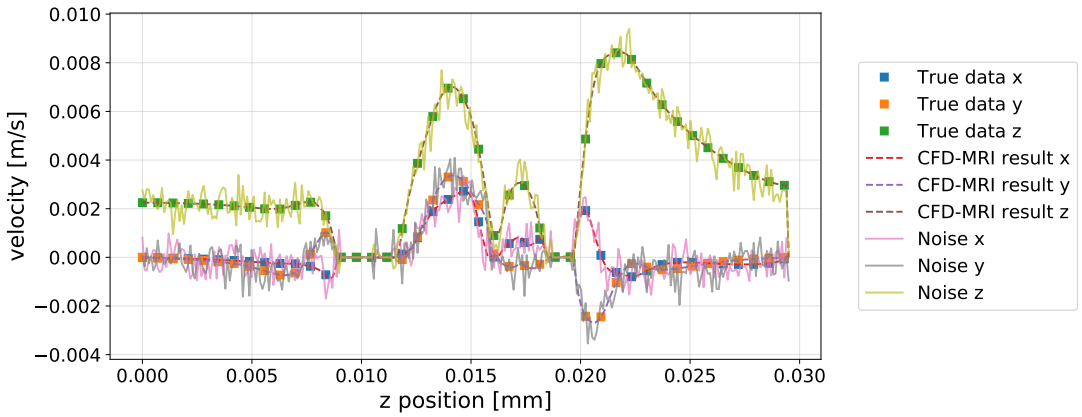
To evaluate the object identification of the porous structure the binary classification is employed as done in Chapter 7. Note that for identification of the object only the noisy velocity field is given, but no information about the object itself. The highest Jaccard index of 86.74% was found at a lattice-porosity threshold of $d_h = 0.98$ (cf. Figure 9.6). At that threshold 96.03% of solid nodes were correctly identified with only 1.91% incorrectly as solid identified nodes, see Table 9.1. The accuracy, which measures correctly identified solid and fluid nodes, is also very high with 97.38%. The precision is still high, but significantly lower with 89.97%. This could be due to the fact that there is no flow at some locations, as may be the case with enclosed cavities in the object. Thus there is no sensitivity for the optimisation and shows the challenge with complex flows. Despite these difficulties, all classification measures are above eighty-six percent (cf. Table 9.2).



(a) Plot along $x-$axis



(b) Plot along $y-$axis



(c) Plot along $z-$axis

Figure 9.5: Velocity plots along a line in the middle of the domain in a) $x-$, b) $y-$, and c) $z-$direction. Shown are all three velocity components of true data, noisy data, and CFD-MRI result.

9.4 Conclusion

In this chapter the CFD-MRI method was applied to a complex flow through a porous structure. The data was the result of a simulation with added noise. The results showed

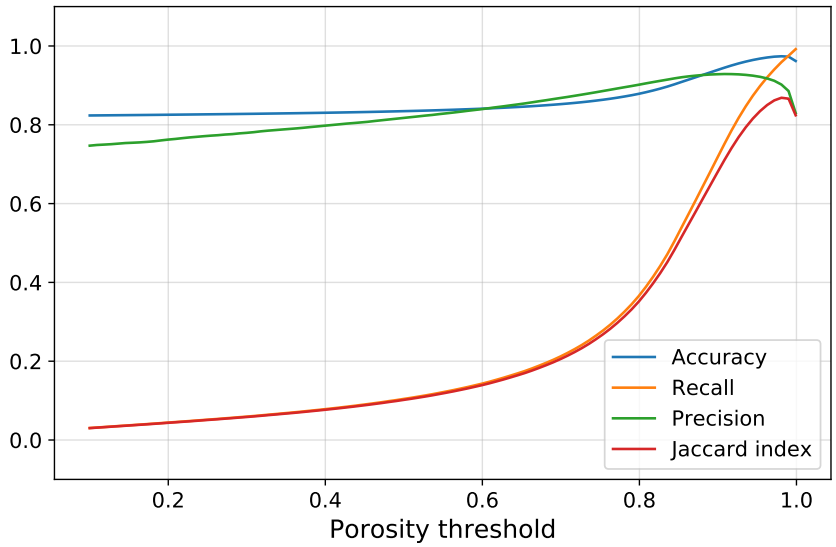


Figure 9.6: Classification measures for different lattice-porosity thresholds.

Outcome	**Nodes**	**Percentage**
True positive	561, 944	96.03% of true solid
True negative	2, 626, 607	97.67% of true fluid
False positive	62, 641	01.91% of total
False negative	23, 233	00.71% of total

Table 9.1: Binary classification outcome for threshold of $d_h = 0.98$.

Measure	**Percentage**
Accuracy	97.38%
Recall	96.03%
Precision	89.97%
Jaccard index	86.74%

Table 9.2: Classification measures for threshold of $d_h = 0.98$.

an accurate characterisation of the true flow, with reduction in the noise and a precise identification of the porous object. The velocity field was found with 4.58% error to the true field, which is a more than eight-fold reduction in error of the noisy field. The porous structure was found with Jaccard index of 86.74% without any knowledge of the object itself, only with the noisy flow field given. These results are very promising as this indicates that expensive and invasive μCT scans could be avoided. Additionally, the use of the adjoint convergence check massively reduced the computation time, and using the Wolfe conditions with the BFGS method increased the stability of the CFD-MRI method for complex applications.

Chapter 10

Summary and outlook

In this thesis the combination of computational fluid dynamics and magnetic resonance imaging measurements, called CFD-MRI, was presented. This combination is formulated as topology optimisation problem with porous media BGK-Boltzmann equation as side condition to minimise the distance from simulated to measured flow field. The problem is discretised using an adjoint lattice Boltzmann method. This combination, its validation, analysis, and application is the mathematical contribution of this thesis.

In a first proof of concept the method was applied to an MRI measurement of a flow around a cylindrical object and indicated that the method was able to find the object in the data and at the same time reduce the measurement noise, even if only very limited information was available

To analyse and validate the method various numerical test cases were constructed. Using simulation data of a flow around a simple, symmetrical and a more complex, asymmetrical object, the results of the object identification were analysed. For this, binary classification measures were used and showed high Jaccard indexes, a measure of similarity, of above 97% for both cases. As expected, if the information in the area of the object was reduced, the Jaccard index was found lower. Also, the simple object was identified more precisely than the complex object. Nonetheless, even if no information of the flow in the object domain was given, the rough location of the object was still found with Jaccard indexes of 69.71% for the simple object and 53.06% for the complex object. It was also found that the total percentage of data was not as important as its location.

In order to analyse the method with regard to noise reduction, synthetic MRI data with low and high noise power was created based on real MRI data. With this the true data, i.e. data without any noise, was available and the CFD-MRI results could be quantified. This showed a significant noise reduction, with an error of less than 1% to the real data

if the full information of the flow field was known, which was a more than eighty-fold reduction in measurement error. The flow information was reduced to only providing velocity data in a plane, and further to the main component of the flow. Even with such limited information available the error was below 1.5% in the given data, and below 10% compared to the whole data, i.e. the three-dimensional domain with all three velocity components. Further, the method was applied to real MRI data, where the high noise image was acquired in thirteen minutes, compared to three hours and twenty-four minutes of the low noise image. Still, the results of the CFD-MRI method for both images differed by less than 2%.

The CFD-MRI method was then applied to a porous structure, where the analysis of noise reduction and object identification done previously were applied to a complex flow that appears in e.g. filtration applications. The porous structure was artificially created based on a real MRI measurement. The result of a flow simulation was then saved and noise was added with 38.55% error to the noise-free data. The results of the CFD-MRI method showed a large decrease in the error to 4.58% with a significant reduction of measurement noise. Further, the complex porous structure was found with a Jaccard index of 86.74% and an accuracy for the domain identification of 97.38%. Additionally, an adjoint convergence check was used to greatly reduce the computation time, and the use of the Wolfe conditions ensured stability of the optimisation.

The presented combination of measurement and simulation shows that a much more precise characterisation of flows and underlying objects and structures can be expected. These improved results promise an extension of the area of application and also make it possible to save costly and limited resources or to improve the overall effectiveness of their use.

In the future, the method could be extended to more complex flow measurements, such as the $4D$ flow MRI, which is capable of measuring time-resolved flows and is often used in medical applications. Checkpointing strategies could be used to cope with the high memory requirements of time-dependent problems. Since the method is based on a model for porous media, the identification of the permeability of homogenised media could also be investigated. The use of regularisation methods, such as Thikonov regularisation, could be of interest for complex applications that may be ill-posed, or to incorporate knowledge about a specific application. Another approach would be to use automatic differentiation to derive the adjoint equations for a variety of different applications, such as thermal flow in porous media.

Bibliography

[1] F. Klemens, S. Schuhmann, G. Guthausen, G. Thäter, M. J. Krause, CFD-MRI: A coupled measurement and simulation approach for accurate fluid flow characterisation and domain identification, Computers & Fluids 166 (2018) 218–224. doi:10.1016/j.compfluid.2018.02.022.

[2] F. Klemens, B. Förster, M. Dorn, G. Thäter, M. J. Krause, Solving fluid flow domain identification problems with adjoint lattice Boltzmann methods, Computers & Mathematics with Applications 79 (1) (2020) 17–33. doi:10.1016/j.camwa.2018.07.010.

[3] F. Klemens, S. Schuhmann, R. Balbierer, G. Guthausen, H. Nirschl, G. Thäter, M. J. Krause, Noise reduction of flow MRI measurements using a lattice Boltzmann based topology optimisation approach, Computers & Fluids 197 (2020) 104391. doi:10.1016/j.compfluid.2019.104391.

[4] B. K. W. Lahoz, R. Menard, Data assimilation, Springer, 2010.

[5] E. Kalnay, Atmospheric modeling, data assimilation and predictability, Cambridge University Press, 2003.

[6] M. D'Elia, L. Mirabella, T. Passerini, M. Perego, M. Piccinelli, C. Vergara, A. Veneziani, Applications of variational data assimilation in computational hemodynamics, in: Modeling of Physiological Flows, Springer, 2012, pp. 363–394. doi:10.1007/978-88-470-1935-5_12.

[7] M. D'Elia, M. Perego, A. Veneziani, A variational data assimilation procedure for the incompressible Navier–Stokes equations in hemodynamics, Journal of Scientific Computing 52 (2) (2012) 340–359. doi:10.1007/s10915-011-9547-6.

[8] T. Guerra, J. Tiago, A. Sequeira, Optimal control in blood flow simulations, International Journal of Non-Linear Mechanics 64 (2014) 57–69. doi:10.1016/j.ijnonlinmec.2014.04.005.

[9] S. W. Funke, M. Nordaas, Ø. Evju, M. S. Alnæs, K. A. Mardal, Variational data assimilation for transient blood flow simulations: Cerebral aneurysms as an illustrative example, International journal for numerical methods in biomedical engineering 35 (1) (2019) e3152. doi:10.1002/cnm.3152.

[10] T. Borrvall, J. Petersson, Topology optimization of fluids in Stokes flow, International Journal for Numerical Methods in Fluids 41 (1) (2003) 77–107. doi:10.1002/fld.426.

[11] A. Gersborg-Hansen, O. Sigmund, R. B. Haber, Topology optimization of channel flow problems, Structural and Multidisciplinary Optimization 30 (3) (2005) 181–192. doi:10.1007/s00158-005-0584-3.

[12] G. Pingen, A. Evgrafov, K. Maute, Topology optimization of flow domains using the lattice Boltzmann method, Structural and Multidisciplinary Optimization 34 (6) (2007) 507–524. doi:10.1007/s00158-007-0105-7.

[13] G. Pingen, A. Evgrafov, K. Maute, A parallel schur complement solver for the solution of the adjoint steady-state lattice Boltzmann equations: application to design optimisation, International Journal of Computational Fluid Dynamics 22 (7) (2008) 457–464. doi:10.1080/10618560802238267.

[14] G. Pingen, M. Waidmann, A. Evgrafov, K. Maute, A parametric level-set approach for topology optimization of flow domains, Structural and Multidisciplinary Optimization 41 (1) (2010) 117–131. doi:10.1007/s00158-009-0405-1.

[15] G. Pingen, K. Maute, Optimal design for non-newtonian flows using a topology optimization approach, Computers & Mathematics with Applications 59 (7) (2010) 2340–2350. doi:10.1016/j.camwa.2009.08.044.

[16] G. Liu, M. Geier, Z. Liu, M. Krafczyk, T. Chen, Discrete adjoint sensitivity analysis for fluid flow topology optimization based on the generalized lattice Boltzmann method, Computers & Mathematics with Applications 68 (10) (2014) 1374–1392. doi:10.1016/j.camwa.2014.09.002.

[17] A. Kirk, S. Kreissl, G. Pingen, K. Maute, Lattice Boltzmann topology optimization for transient flow, in: MAESC 2011 Conference May, Vol. 3, 2011.

[18] S. Kreissl, G. Pingen, K. Maute, Topology optimization for unsteady flow, International Journal for Numerical Methods in Engineering 87 (13) (2011) 1229–1253. doi:10.1002/nme.3151.

[19] S. Nørgaard, O. Sigmund, B. Lazarov, Topology optimization of unsteady flow problems using the lattice Boltzmann method, Journal of Computational Physics 307 (2016) 291–307. doi:10.1016/j.jcp.2015.12.023.

[20] C. Chen, K. Yaji, T. Yamada, K. Izui, S. Nishiwaki, Local-in-time adjoint-based topology optimization of unsteady fluid flows using the lattice Boltzmann method, Mechanical Engineering Journal 4 (3) (2017) 17–00120. doi:10.1299/mej.17-00120.

[21] S. Kreissl, G. Pingen, A. Evgrafov, K. Maute, Topology optimization of flexible microfluidic devices, Structural and Multidisciplinary Optimization 42 (4) (2010) 495–516. doi:10.1007/s00158-010-0526-6.

[22] D. Makhija, G. Pingen, R. Yang, K. Maute, Topology optimization of multi-component flows using a multi-relaxation time lattice Boltzmann method, Computers & Fluids 67 (2012) 104–114. doi:10.1016/j.compfluid.2012.06.018.

[23] Ł. Łaniewski Wołłk, J. Rokicki, Adjoint lattice Boltzmann for topology optimization on multi-GPU architecture, Computers & Mathematics with Applications 71 (3) (2016) 833–848. doi:10.1016/j.camwa.2015.12.043.

[24] K. Yaji, T. Yamada, M. Yoshino, T. Matsumoto, K. Izui, S. Nishiwaki, Topology optimization using the lattice Boltzmann method incorporating level set boundary expressions, Journal of Computational Physics 274 (2014) 158–181. doi:10.1016/j.jcp.2014.06.004.

[25] F. Dugast, Y. Favennec, C. Josset, Y. Fan, L. Luo, Topology optimization of thermal fluid flows with an adjoint lattice Boltzmann method, Journal of Computational Physics 365 (2018) 376–404. doi:10.1016/j.jcp.2018.03.040.

[26] M. M. Tekitek, M. Bouzidi, F. Dubois, P. Lallemand, Adjoint lattice Boltzmann equation for parameter identification, Computers & Fluids 35 (2006) 805–813. doi:10.1016/j.compfluid.2005.07.015.

[27] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with PDE constraints, Vol. 23, Springer Science & Business Media, 2008. doi:10.1007/978-1-4020-8839-1.

[28] M. D. Gunzburger, Perspectives in flow control and optimization, Vol. 5, Siam, 2003.

[29] M. J. Krause, G. Thäter, V. Heuveline, Adjoint-based fluid flow control and optimisation with lattice Boltzmann methods, Computers & Mathematics with Applications 65 (6) (2013) 945–960, Mesoscopic Methods in Engineering and Science. doi:10.1016/j.camwa.2012.08.007.

[30] M. J. Krause, B. Förster, A. Mink, H. Nirschl, Towards solving fluid flow domain identification problems with adjoint lattice Boltzmann methods, in: High Performance Computing in Science and Engineering '16, Springer International Publishing, Cham, 2016, pp. 337–353. doi:10.1007/978-3-319-47066-5_23.

[31] P. Wesseling, Principles of Computational Fluid Dynamics, Springer-Verlag Berlin Heidelberg, 2001. doi:10.1007/978-3-642-05146-3.

[32] M. Feistauer, J. Felcman, I. Straŝkraba, Mathematical and Computational Methods for Compressible Flow, Oxford University Press, 2003.

[33] J. H. Ferziger, M. Peric, Numerische Strömungsmechanik, Springer-Verlag Berlin Heidelberg, 2008. doi:10.1007/978-3-540-68228-8.

[34] R. K. Zeytounian, Challenges in Fluid Dynamics: A New Approach, Springer International Publishing, 2017. doi:10.1007/978-3-319-31619-2.

[35] G. Łukaszewicz, P. Kalita, Navier–Stokes Equations: An Introduction with Applications, Springer International Publishing, 2016. doi:10.1007/978-3-319-27760-8.

[36] S. Jayanti, Computational Fluid Dynamics for Engineers and Scientists, Springer Netherlands, 2018. doi:10.1007/978-94-024-1217-8.

[37] M. Feistauer, Mathematical Methods in Fluid Dynamics, Longman Scientific & Technical, Harlow, England, 1993.

[38] C. L. Fefferman, Existence and smoothness of the Navier–Stokes equation, The millennium prize problems 57 (2006) 67.

[39] H. P. G. Darcy, Les Fontaines publiques de la ville de Dijon. Exposition et application des principes à suivre et des formules à employer dans les questions de distribution d'eau, etc, Dalamont, Paris, 1856.

[40] S. Whitaker, The method of volume averaging, Vol. 13, Springer Netherlands, 2013. doi:10.1007/978-94-017-3389-2.

[41] S. Whitaker, Flow in porous media I: A theoretical derivation of Darcy's law, Transport in porous media 1 (1) (1986) 3–25. doi:10.1007/BF01036523.

[42] A. Bensoussan, J.-L. Lions, G. Papanicolau, Asymptotic analysis for periodic structures, North Holland, Amsterdam, 1978.

[43] E. Sanchez-Palencia, Non Homogeneous Media and Vibration Theory, Vol. 127 of Lecture Notes in Physics, Springer, Berlin, 1980. doi:10.1007/3-540-10000-8.

[44] G. Allaire, Homogenization of the Navier–Stokes equations in open sets perforated with tiny holes I. Abstract framework, a volume distribution of holes, Archive for Rational Mechanics and Analysis 113 (3) (1991) 209–259. doi:10.1007/BF00375065.

[45] G. Allaire, Homogenization of the Navier–Stokes equations in open sets perforated with tiny holes II: Non-critical sizes of the holes for a volume distribution and a surface distribution of holes, Archive for Rational Mechanics and Analysis 113 (3) (1991) 261–298. doi:10.1007/BF00375066.

[46] H. Brinkman, A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles, Flow, Turbulence and Combustion 1 (1) (1949) 27. doi:10.1007/BF02120313.

[47] H. Brinkman, On the permeability of media consisting of closely packed porous particles, Flow, Turbulence and Combustion 1 (1) (1949) 81. doi:10.1007/BF02120318.

[48] N. S. Martys, Improved approximation of the Brinkman equation using a lattice Boltzmann method, Physics of Fluids 13 (6) (2001) 1807–1810. doi:10.1063/1.1368846.

[49] D. A. Nield, A. Bejan, Convection in Porous Media, Springer, 2017. doi:10.1007/978-3-319-49562-0.

[50] A. Mikelić, Homogenization of nonstationary Navier–Stokes equations in a domain with a grained boundary, Annali di Matematica pura ed applicata 158 (1) (1991) 167–179.

[51] D. D. Joseph, D. A. Nield, G. Papanicolaou, Nonlinear equation governing flow in a saturated porous medium, Water Resources Research 18 (4) (1982) 1049–1052. doi:10.1029/WR018i004p01049.

[52] S. Ergun, Fluid flow through packed columns, Chemical Engineering Progress 48 (1952) 89–94.

[53] K. Vafai, Convective flow and heat transfer in variable-porosity media, Journal of Fluid Mechanics 147 (1984) 233–259. doi:10.1017/S002211208400207X.

[54] C. Cercignani, Rarefied Gas Dynamics: From Basic Concepts to Actual Calculations, Cambridge University Press, 2000.

[55] D. Hänel, Molekulare Gasdynamik: Einführung in die kinetische Theorie der Gase und Lattice-Boltzmann-Methoden, Springer, 2004. doi:10.1007/3-540-35047-0.

[56] H. Babovsky, Die Boltzmann-Gleichung: Modellbildung-Numerik-Anwendungen, Vieweg+Teubner Verlag, 1998. doi:10.1007/978-3-663-12034-6.

[57] L. Saint-Raymond, Hydrodynamic Limits of the Boltzmann Equation, Springer, 2009. doi:10.1007/978-3-540-92847-8.

[58] C. Villani, A review of mathematical topics in collisional kinetic theory, Handbook of Mathematical Fluid Dynamics 1 (71-305) (2002) 3–8.

[59] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E. M. Viggen, The Lattice Boltzmann Method: Principles and Practice, Graduate Texts in Physics, Springer, 2017. doi:10.1007/978-3-319-44649-3.

[60] C. Cercignani, The Boltzmann Equation and Its Applications, Springer-Verlag, New York, 1988.

[61] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Physical Review 94 (3) (1954) 511. doi:10.1103/PhysRev.94.511.

[62] S. Chapman, T. G. Cowling, The Mathematical Theory of Non-uniform Gases, Cambridge University Press, 1952.

[63] L. Saint-Raymond, From the BGK model to the Navier–Stokes equations, Annales scientifiques de l'École Normale Supérieure Ser. 4, 36 (2) (2003) 271–317. doi:10.1016/S0012-9593(03)00010-7.

[64] B. Perthame, Global existence to the BGK model of Boltzmann equation, Journal of Differential Equations 82 (1) (1989) 191–205. doi:10.1016/0022-0396(89)90173-3.

[65] B. Perthame, A. P. N. Dinh, The Dirichlet boundary value problem for BGK equation, in: Advances in Kinetic Theory and Continuum Mechanics, Springer, 1991, pp. 13–18. doi:10.1007/978-3-642-50235-4_2.

[66] M. C. Sukop, D. T. Thorne, Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers, Springer, 2006. doi:10.1007/978-3-540-27982-2.

[67] Z. Guo, C. Shu, Lattice Boltzmann Method and Its Applications in Engineering, World Scientific, 2013. doi:10.1142/8806.

[68] M. Gaedtke, S. Wachter, M. Rädle, H. Nirschl, M. J. Krause, Application of a lattice Boltzmann method combined with a Smagorinsky turbulence model to spatially resolved heat flux inside a refrigerated vehicle, Computers & Mathematics with Applications 76 (10) (2018) 2315–2329. doi:10.1016/j.camwa.2018.08.018.

[69] M. Gaedtke, T. Hoffmann, V. Reinhardt, G. Thäter, H. Nirschl, M. J. Krause, Flow and heat transfer simulation with a thermal large eddy lattice Boltzmann method in an annular gap with an inner rotating cylinder, International Journal of Modern Physics C (IJMPC) 30 (02n03) (2019) 1–25. doi:10.1142/S012918311950013X.

[70] M. A. A. Spaid, F. R. Phelan Jr., Lattice Boltzmann methods for modeling microscale flow in fibrous porous media, Physics of Fluids 9 (9) (1997) 2468–2474. doi:10.1063/1.869392.

[71] Z. Guo, T. S. Zhao, Lattice Boltzmann model for incompressible flows through porous media, Phys. Rev. E 66 (2002) 036304. doi:10.1103/PhysRevE.66.036304.

[72] J. Ross-Jones, M. Gaedtke, S. Sonnick, M. Rädle, H. Nirschl, M. J. Krause, Conjugate heat transfer through nano scale porous media to optimize vacuum insulation panels with lattice Boltzmann methods, Computers & Mathematics with Applications 77 (1) (2019) 209–221. doi:10.1016/j.camwa.2018.09.023.

[73] M. Haussmann, S. Simonis, H. Nirschl, M. J. Krause, Direct numerical simulation of decaying homogeneous isotropic turbulence — numerical experiments on stability, consistency and accuracy of distinct lattice Boltzmann methods, International Journal of Modern Physics C (IJMPC) 30 (09) (2019) 1–29. doi:10.1142/S0129183119500748.

[74] M. Haussmann, A. C. BARRETO, G. L. KOUYI, N. Rivière, H. Nirschl, M. J. Krause, Large-eddy simulation coupled with wall models for turbulent channel flows at high Reynolds numbers with a lattice Boltzmann method — Application to Coriolis mass flowmeter, Computers & Mathematics with Applications 78 (10) (2019) 3285–3302. doi:10.1016/j.camwa.2019.04.033.

[75] R. Trunk, T. Henn, W. Dörfler, H. Nirschl, M. J. Krause, Inertial dilute particulate fluid flow simulations with an Euler–Euler lattice Boltzmann method, Journal of Computational Science 17 (2016) 438–445, discrete Simulation of Fluid Dynamics 2015. doi:10.1016/j.jocs.2016.03.013.

[76] M. J. Krause, F. Klemens, T. Henn, R. Trunk, H. Nirschl, Particle flow simulations with homogenised lattice Boltzmann methods, Particuology 34 (2017) 1–13. doi:10.1016/j.partic.2016.11.001.

[77] R. Trunk, J. Marquardt, G. Thäter, H. Nirschl, M. J. Krause, Towards the simulation of arbitrarily shaped 3D particles using a homogenised lattice Boltzmann method, Computers & Fluids 172 (2018) 621–631. doi:10.1016/j.compfluid.2018.02.027.

[78] E. M. Viggen, Viscously damped acoustic waves with the lattice Boltzmann method, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 369 (1944) (2011) 2246–2254. doi:10.1098/rsta.2011.0040.

[79] A. Mink, G. Thäter, H. Nirschl, M. J. Krause, A 3D lattice Boltzmann method for light simulation in participating media, Journal of Computational Science 17 (2016) 431–437, discrete Simulation of Fluid Dynamics 2015. doi:10.1016/j.jocs.2016.03.014.

[80] A. Mink, C. McHardy, L. Bressel, C. Rauh, M. J. Krause, Radiative transfer lattice Boltzmann methods: 3D models and their performance in different regimes of radiative transfer, Journal of Quantitative Spectroscopy and Radiative Transfer 243 (2020) 106810. doi:10.1016/j.jqsrt.2019.106810.

[81] D. A. Wolf-Gladrow, Lattice Gas Cellular Automata and Lattice Boltzmann Models, Springer, New York, 2005.

[82] X. He, L.-S. Luo, Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, Phys. Rev. E 56 (1997) 6811–6817. doi:10.1103/PhysRevE.56.6811.

[83] X. Shan, X.-F. Yuan, H. Chen, Kinetic theory representation of hydrodynamics: A way beyond the Navier—Stokes equation, Journal of Fluid Mechanics 550 (2006) 413–441. doi:10.1017/S0022112005008153.

[84] P. Davis, P. Rabinowitz, Methods of Numerical Integration, Dover Books on Mathematics Series, Dover Publications, 2007.

[85] S. Ubertini, P. Asinari, S. Succi, Three ways to lattice Boltzmann: A unified time-marching picture, Phys. Rev. E 81 (2010) 016311. doi:10.1103/PhysRevE.81.016311.

[86] T. Pohl, F. Deserno, N. Thurey, U. Rude, P. Lammers, G. Wellein, T. Zeiser, Performance evaluation of parallel large-scale lattice Boltzmann applications on three supercomputing architectures, in: SC '04: Proceedings of the 2004 ACM/IEEE Conference on Supercomputing, 2004, pp. 21–21. doi:10.1109/SC.2004.37.

[87] C. Körner, T. Pohl, U. Rüde, N. Thürey, T. Zeiser, Parallel Lattice Boltzmann Methods for CFD Applications, in: A. M. Bruaset, A. Tveito (Eds.), Numerical Solution of Partial Differential Equations on Parallel Computers, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 439–466.

[88] A. J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation, Journal of Fluid Mechanics 271 (1994) 285—-309. doi:10.1017/S0022112094001771.

[89] M. C. Sukop, D. Or, Lattice Boltzmann method for modeling liquid-vapor interface configurations in porous media, Water Resources Research 40 (1). doi:10.1029/2003WR002333.

[90] I. Ginzbourg, D. d'Humières, Local second-order boundary methods for lattice Boltzmann models, Journal of Statistical Physics 84 (5) (1996) 927–971. doi:10.1007/BF02174124.

[91] M. Bouzidi, M. Firdaouss, P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, Physics of Fluids 13 (11) (2001) 3452–3459. doi:10.1063/1.1399290.

[92] Q. Zou, X. He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, Physics of Fluids 9 (6) (1997) 1591–1598. doi:10.1063/1.869307.

[93] H. Huang, M. Krafczyk, X. Lu, Forcing term in single-phase and Shan-Chen-type multiphase lattice Boltzmann models, Phys. Rev. E 84 (2011) 046710. doi:10.1103/PhysRevE.84.046710.

[94] X. Shan, H. Chen, Lattice Boltzmann model for simulating flows with multiple phases and components, Phys. Rev. E 47 (1993) 1815–1819. doi:10.1103/PhysRevE.47.1815.

[95] Z. Guo, C. Zheng, B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, Phys. Rev. E 65 (2002) 046308. doi:10.1103/PhysRevE.65.046308.

[96] Modeling void formation dynamics in fibrous porous media with the lattice Boltzmann method, Composites Part A: Applied Science and Manufacturing 29 (7) (1998) 749–755. doi:10.1016/S1359-835X(98)00031-1.

[97] J. Olivares, P. Martin, E. Valero, A simple approximation for the modified Bessel function of zero order I0(x), in: Journal of Physics: Conference Series, Vol. 1043, IOP Publishing, 2018, p. 012003. doi:10.1088/1742-6596/1043/1/012003.

[98] P. T. Callaghan, Principles of Nuclear Magnetic Resonance Microscopy, Oxford University Press, New York, 1991.

[99] M. F. Reiser, W. Semmler, H. Hricak, Magnetic Resonance Tomography, Springer-Verlag Berlin Heidelberg, 2008. doi:10.1007/978-3-540-29355-2.

[100] C. J. Elkins, M. T. Alley, Magnetic resonance velocimetry: Applications of magnetic resonance imaging in the measurement of fluid motion, Experiments in Fluids 43 (6) (2007) 823–858. doi:10.1007/s00348-007-0383-2.

[101] Z. Stankovic, B. D. Allen, J. Garcia, K. B. Jarvis, M. Markl, 4D flow imaging with MRI, Cardiovascular Diagnosis and Therapy 4 (2) (2014) 173. doi:10.3978/j.issn.2223-3652.2014.01.02.

[102] L. F. Gladden, Magnetic resonance: Ongoing and future role in chemical engineering research, AIChE Journal 49 (1) (2003) 2–9. doi:10.1002/aic.690490102.

[103] J. D. Seymour, P. T. Callaghan, Generalized approach to NMR analysis of flow and dispersion in porous media, AIChE Journal 43 (8) (1997) 2096–2111. doi:10.1002/aic.690430817.

[104] B. Manz, L. F. Gladden, P. B. Warren, Flow and dispersion in porous media: Lattice-Boltzmann and NMR studies, AIChE Journal 45 (9) (1999) 1845–1854. doi:10.1002/aic.690450902.

[105] P. R. Moran, A flow velocity zeugmatographic interlace for NMR imaging in humans, Magnetic Resonance Imaging 1 (4) (1982) 197–203, second Annual Meeting of the Society for Magnetic Resonance Imaging. doi:10.1016/0730-725X(82)90170-9.

[106] D. Bryant, J. Payne, D. Firmin, D. Longmore, Measurement of flow with NMR imaging using a gradient pulse and phase difference technique, J Comput Assist Tomogr 8 (4) (1984) 588–593. doi:10.1097/00004728-198408000-00002.

[107] N. J. Pelc, R. J. Herfkens, A. Shimakawa, D. R. Enzmann, et al., Phase contrast cine magnetic resonance imaging, Magnetic Resonance Quarterly 7 (4) (1991) 229–254.

[108] C. A. Taylor, M. T. Draney, Experimental and computational methods in cardiovascular fluid mechanics, Annual Review of Fluid Mechanics 36 (1) (2004) 197–231. doi:10.1146/annurev.fluid.36.050802.121944.

[109] J. Lotz, C. Meier, A. Leppert, M. Galanski, Cardiovascular flow measurement with phase-contrast MR imaging: Basic facts and implementation, Radiographics 22 (3) (2002) 651–671. doi:10.1148/radiographics.22.3.g02ma11651.

[110] J. Mohan, V. Krishnaveni, Y. Guo, A survey on the magnetic resonance image denoising methods, Biomedical Signal Processing and Control 9 (2014) 56–69. doi:10.1016/j.bspc.2013.10.007.

[111] H. Zhu, Y. Li, J. G. Ibrahim, X. Shi, H. An, Y. Chen, W. Gao, W. Lin, D. B. Rowe, B. S. Peterson, Regression models for identifying noise sources in magnetic resonance images, Journal of the American Statistical Association 104 (486) (2009) 623–637. doi:10.1198/jasa.2009.0029.

[112] H. Gudbjartsson, S. Patz, The Rician distribution of noisy MRI data, Magnetic Resonance in Medicine 34 (6) (1995) 910–914.

[113] A. Macovski, Noise in MRI, Magnetic Resonance in Medicine 36 (3) (1996) 494–497. doi:10.1002/mrm.1910360327.

[114] D. B. Rowe, B. R. Logan, A complex way to compute fMRI activation, NeuroImage 23 (3) (2004) 1078–1092. doi:10.1016/j.neuroimage.2004.06.042.

[115] R. M. Henkelman, Measurement of signal intensities in the presence of noise in MR images, Medical Physics 12 (2) (1985) 232–233. doi:10.1118/1.595711.

[116] S. Schuhmann, N. Schork, K. Beller, H. Nirschl, T. Oerther, G. Guthausen, In-situ characterization of deposits in ceramic hollow fiber membranes by compressed sensing RARE-MRI, AIChE Journal 64 (11) (2018) 4039–4046. doi:10.1002/aic.16201.

[117] A. Andersen, J. Kirsch, Analysis of noise in phase contrast MR imaging, Medical Physics 23 (6) (1996) 857–869. doi:10.1118/1.597729.

[118] F. Tröltzsch, Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen, Springer-Verlag, 2009. doi:10.1007/978-3-8348-9357-4.

[119] J. C. De los Reyes, Numerical PDE-constrained optimization, Springer, 2015. doi:10.1007/978-3-319-13395-9.

[120] J. Nocedal, S. Wright, Numerical optimization, Springer Science & Business Media, 2006. doi:10.1007/b98874.

[121] P. Wolfe, Convergence conditions for ascent methods, SIAM Review 11 (2) (1969) 226–235. doi:10.1137/1011036.

[122] P. Wolfe, Convergence conditions for ascent methods. II: Some corrections, SIAM Review 13 (2) (1971) 185–188. doi:10.1137/1013035.

[123] C. G. Broyden, The convergence of a class of double-rank minimization algorithms 1. General considerations, IMA Journal of Applied Mathematics 6 (1) (1970) 76–90. doi:10.1093/imamat/6.1.76.

[124] C. G. Broyden, The convergence of a class of double-rank minimization algorithms: 2. The new algorithm, IMA journal of applied mathematics 6 (3) (1970) 222–231. doi:10.1093/imamat/6.3.222.

[125] R. Fletcher, A new approach to variable metric algorithms, The Computer Journal 13 (3) (1970) 317–322. doi:10.1093/comjnl/13.3.317.

[126] D. Goldfarb, A family of variable-metric methods derived by variational means, Mathematics of Computation 24 (109) (1970) 23–26. doi:10.1090/S0025-5718-1970-0258249-6.

[127] D. F. Shanno, Conditioning of quasi-Newton methods for function minimization, Mathematics of Computation 24 (111) (1970) 647–656. doi:doi.org/10.1090/S0025-5718-1970-0274029-X.

[128] W. Sun, Y.-X. Yuan, Optimization Theory and Methods: Nonlinear Programming, Vol. 1, Springer Science & Business Media, 2006.

[129] R. H. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM Journal on Scientific Computing 16 (5) (1995) 1190–1208. doi:10.1137/0916069.

[130] H. H. Rosenbrock, An Automatic Method for Finding the Greatest or Least Value of a Function, The Computer Journal 3 (3) (1960) 175–184. doi:10.1093/comjnl/3.3.175.

[131] L. Dixon, D. Mills, Effect of rounding errors on the variable metric method, Journal of Optimization Theory and Applications 80 (1) (1994) 175–179. doi:10.1007/BF02196600.

[132] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: A survey, Journal of Machine Learning Research 18 (153) (2018) 1–43.

[133] A. Griewank, A. Walther, Evaluating derivatives: principles and techniques of algorithmic differentiation, Vol. 105, Siam, 2008. doi:10.1137/1.9780898717761.

[134] A. Griewank, A. Walther, Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation, ACM Transactions on Mathematical Software (TOMS) 26 (1) (2000) 19–45. doi:10.1145/347837.347846.

[135] C. H. Bischof, P. D. Hovland, B. Norris, On the implementation of automatic differentiation tools, Higher-Order and Symbolic Computation 21 (3) (2008) 311–331. doi:10.1007/s10990-008-9034-4.

[136] M. J. Krause, Fluid flow simulation and optimisation with lattice Boltzmann methods on high performance computers - application to the human respiratory system, Ph.D. thesis, Karlsruhe Institute of Technology (KIT) (2010).

[137] M. J. Krause, V. Heuveline, Parallel fluid flow control and optimisation with lattice Boltzmann methods and automatic differentiation, Computers & Fluids 80 (2013) 28–36. doi:10.1016/j.compfluid.2012.07.026.

[138] E. Casas, R. Herzog, G. Wachsmuth, Optimality conditions and error analysis of semilinear elliptic control problems with L1 cost functional, SIAM Journal on Optimization 22 (3) (2012) 795–820. doi:10.1137/110834366.

[139] V. Heuveline, M. Krause, J. Latt, Towards a hybrid parallelization of lattice Boltzmann methods, Computers & Mathematics with Applications 58 (2009) 1071–1080. doi:10.1016/j.camwa.2009.04.001.

[140] K. Yonekura, Y. Kanno, A flow topology optimization method for steady state flow using transient information of flow field solved by lattice Boltzmann method, Structural and Multidisciplinary Optimization 51 (1) (2015) 159–172. doi:10.1007/s00158-014-1123-x.

[141] K. Yaji, T. Yamada, M. Yoshino, T. Matsumoto, K. Izui, S. Nishiwaki, Topology optimization in thermal-fluid flow using the lattice Boltzmann method, Journal of Computational Physics 307 (2016) 355–377. doi:10.1016/j.jcp.2015.12.008.

[142] B. Karlik, A. V. Olgac, Performance analysis of various activation functions in generalized MLP architectures of neural networks, International Journal of Artificial Intelligence and Expert Systems 1 (4) (2011) 111–122.

[143] G. Turk, M. Levoy, The Stanford bunny. the Stanford 3d scanning repository (1994). URL http://www-graphics.stanford.edu/data/3Dscanrep

[144] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874, rOC Analysis in Pattern Recognition. doi:10.1016/j.patrec.2005.10.010.

[145] E. A. Freeman, G. G. Moisen, A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa, Ecological Modelling 217 (1) (2008) 48–58. doi:10.1016/j.ecolmodel.2008.05.015.

[146] D. Powers, Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation, J. Mach. Learn. Technol 2 (1) (2011) 37–63.

[147] P. Jaccard, Lois de distribution florale dans la zone alpine, Bull Soc Vaudoise Sci Nat 38 (1902) 69–130.

[148] M. Levandowsky, D. Winter, Distance between sets, Nature 234 (5323) (1971) 34–35. doi:10.1038/234034a0.

[149] R. Shi, K. N. Ngan, S. Li, Jaccard index compensation for object segmentation evaluation, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 4457–4461.

[150] R. M. Cronin, D. Fabbri, J. C. Denny, G. P. Jackson, Automated classification of consumer health information needs in patient portal messages, in: AMIA Annual Symposium Proceedings, Vol. 2015, 2015, p. 1861.

[151] E. R. McVeigh, R. M. Henkelman, M. J. Bronskill, Noise and filtration in magnetic resonance imaging, Medical Physics 12 (5) (1985) 586–591. doi:10.1118/1.595679.

[152] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7) (1990) 629–639. doi:10.1109/34.56205.

[153] A. Buades, B. Coll, J. M. Morel, A review of image denoising algorithms, with a new one, Multiscale Modeling & Simulation 4 (2) (2005) 490–530. doi:10.1137/040616024.

[154] J. B. Weaver, Y. Xu, D. M. Healy Jr., L. D. Cromwell, Filtering noise from images with wavelet transforms, Magnetic Resonance in Medicine 21 (2) (1991) 288–295. doi:10.1002/mrm.1910210213.

[155] J.-L. Starck, E. J. Candes, D. L. Donoho, The curvelet transform for image denoising, IEEE Transactions on Image Processing 11 (6) (2002) 670–684. doi:10.1109/TIP.2002.1014998.

[156] L. He, I. R. Greenshields, A nonlocal maximum likelihood estimation method for Rician noise reduction in MR images, IEEE Transactions on Medical Imaging 28 (2) (2009) 165–172. doi:10.1109/TMI.2008.927338.

[157] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, IEEE Transactions on Image Processing 26 (7) (2017) 3142–3155. doi:10.1109/TIP.2017.2662206.

[158] F. Ong, M. Uecker, U. Tariq, A. Hsiao, M. T. Alley, S. S. Vasanawala, M. Lustig, Robust 4D flow denoising using divergence-free wavelet transform, Magnetic Resonance in Medicine 73 (2) (2015) 828–842. doi:10.1002/mrm.25176.

[159] E. Bostan, S. Lefkimmiatis, O. Vardoulis, N. Stergiopulos, M. Unser, Improved variational denoising of flow fields with application to phase-contrast MRI data, IEEE Signal Processing Letters 22 (6) (2014) 762–766. doi:10.1109/LSP.2014.2369212.

[160] A. Bakhshinejad, A. Baghaie, A. Vali, D. Saloner, V. L. Rayz, R. M. D'Souza, Merging computational fluid dynamics and 4D flow MRI using proper orthogonal decomposition and ridge regression, Journal of biomechanics 58 (2017) 162–173. doi:10.1016/j.jbiomech.2017.05.004.

[161] M. F. Fathi, A. Bakhshinejad, A. Baghaie, D. Saloner, R. H. Sacho, V. L. Rayz, R. M. D'Souza, Denoising and spatial resolution enhancement of 4D flow MRI using proper orthogonal decomposition and lasso regularization, Computerized Medical Imaging and Graphics 70 (2018) 165–172. doi:10.1016/j.compmedimag.2018.07.003.

[162] V. C. Rispoli, J. F. Nielsen, K. S. Nayak, J. L. Carvalho, Computational fluid dynamics simulations of blood flow regularized by 3D phase contrast MRI, Biomedical Engineering Online 14 (1) (2015) 110. doi:10.1186/s12938-015-0104-7.

[163] C. Teepakorn, D. Grenier, K. Fiaty, C. Charcosset, Characterization of hydrodynamics in membrane chromatography devices using magnetic resonance imaging and computational fluid dynamics, Chemical Engineering Research and Design 113 (2016) 61–73. doi:10.1016/j.cherd.2016.06.027.

[164] L. Goubergrits, R. Mevert, P. Yevtushenko, J. Schaller, U. Kertzscher, S. Meier, S. Schubert, E. Riesenkampff, T. Kuehne, The impact of MRI-based inflow for the hemodynamic evaluation of aortic coarctation, Annals of Biomedical Engineering 41 (12) (2013) 2575–2587. doi:10.1007/s10439-013-0879-2.

[165] L. Goubergrits, E. Riesenkampff, P. Yevtushenko, J. Schaller, U. Kertzscher, A. Hennemuth, F. Berger, S. Schubert, T. Kuehne, MRI-based computational fluid dynamics for diagnosis and treatment prediction: Clinical validation study in patients with coarctation of aorta, Journal of Magnetic Resonance Imaging 41 (4) (2015) 909–916. doi:10.1002/jmri.24639.

[166] T. Schenkel, M. Malve, M. Reik, M. Markl, B. Jung, H. Oertel, MRI-based CFD analysis of flow in a human left ventricle: Methodology and application to a healthy heart, Annals of Biomedical Engineering 37 (3) (2009) 503–515. doi:10.1007/s10439-008-9627-4.

[167] N. J. Pelc, M. Drangova, L. R. Pelc, Y. Zhu, D. C. Noll, B. S. Bowman, R. J. Herfkens, Tracking of cyclic motion with phase-contrast cine MR velocity data, Journal of Magnetic Resonance Imaging 5 (3) (1995) 339–345. doi:10.1002/jmri.1880050319.

[168] E. R. Jensen, D. A. Morrow, J. P. Felmlee, G. M. Odegard, K. R. Kaufman, Error analysis of cine phase contrast MRI velocity measurements used for strain calculation, Journal of Biomechanics 48 (1) (2015) 95–103. doi:10.1016/j.jbiomech.2014.10.035.

Appendix A

Publications

The articles [1–3] have been published in the last years and form the basis for parts of this thesis. They are listed below, including a description where they are included in this work, and the contributions of each author based on the *CRediT*[1] statements.

CFD-MRI: A coupled measurement and simulation approach for accurate fluid flow characterisation and domain identification
Fabian Klemens, Sebastian Schuhmann, Gisela Guthausen, Gudrun Thäter, Mathias J. Krause

Abstract
This article presents the coupling of magnetic resonance imaging (MRI) measurements and computational fluid dynamics (CFD) for accurate characterisation of fluid flow and identification of flow domains. Currently, MRI measurements are averaged over time and space, assuming a certain smoothness of the velocity and pressure space. However, a possible solution of a fluid problem must fulfil the Navier–Stokes equations, which sets up a condition that is much more restrictive than the usual smoothness assumptions in e.g. curve fitting. The novel CFD-MRI method uses this insight to reduce the statistical noise and to identify finer structures of the underlying domain. The problem is formulated as a distributed control problem which minimises the distance between measured and simulated flow field. Thereby, the simulated flow field is the solution of a parametrised porous media BGK-Boltzmann equation which approaches a homogenised Navier–Stokes equation in the hydrodynamic limit. The parameters represent the porosity distributed in the domain which yields a domain and a fluid flow that fits best to the measured data. This enables the method they locate an obstacle and the flow field from limited $2D$ spatially resolved MRI data with one velocity component. The problem is solved with an adjoint lattice Boltzmann method (ALBM) using the open source software OpenLB.

Contribution
Fabian Klemens: Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing. **Sebastian Schuhmann**: Investigation, Resources, Data curation. **Gisela Guthausen**: Supervision, Project administration, Funding acquisition. **Gudrun Thäter**: Supervision. **Mathias J. Krause**: Supervision, Project administration, Funding acquisition, Software, Writing - review & editing.

[1] `https://www.elsevier.com/authors/journal-authors/policies-and-ethics/credit-author-statement`

Comment
This paper partially forms Chapter 6.

Solving fluid flow domain identification problems with adjoint lattice Boltzmann methods
Fabian Klemens, Benjamin Förster, Márcio Dorn, Gudrun Thäter, Mathias J. Krause

Abstract
In this article, the adjoint lattice Boltzmann method (ALBM) for solving fluid domain identification problems for incompressible fluids, proposed by Krause et al. (2016), is improved and validated. The problem is formulated as a distributed control problem which minimises the distance between a given, e.g. from measurements like MRI, and a simulated flow field. Thereby, the simulated flow field is the solution of a parametrised porous media BGK–Boltzmann problem, where the parameters represent porosity distributed in the domain. The proposed parametrisation consists of linking the variables representing a lattice-dependent porosity with the control variables. Hereby, it is paid attention that a given control parameter set yields results which are independent of the underlying grid resolution. It enables solving an optimisation problem with different resolutions without adapting the initial set of control variables.

Contribution
Fabian Klemens: Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing. **Benjamin Förster**: Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft. **Márcio Dorn**: Supervision. **Gudrun Thäter**: Supervision. **Mathias J. Krause**: Conceptualisation, Methodology, Supervision, Project administration, Funding acquisition, Software, Writing - review & editing.

Comment
This paper partially forms Section 5.6 and Chapter 7.

Noise reduction of flow MRI measurements using a lattice Boltzmann based topology optimisation approach
Fabian Klemens, Sebastian Schuhmann, Roland Balbierer, Gisela Guthausen, Hermann Nirschl, Gudrun Thäter, Mathias J. Krause

Abstract
In a previous work, the feasibility of coupling magnetic resonance imaging (MRI) measurements and computational fluid dynamics (CFD) was presented, called CFD-MRI. Using a lattice Boltzmann based topology optimisation approach, the method can be described as a Navier–Stokes filter for flow MRI measurements. The main objective of this article is the analysis and quantification of CFD-MRI for its ability to reduce statistical measurement noise. For this, MRI data was analysed and used as basis for synthetic data, where noise was added to a simulation result. Thus, the noise-free data is known and a thorough analysis can be performed. The results show a very high agreement with the original data, even with high statistical noise in the input data and limited information available.

Contribution
Fabian Klemens: Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing. **Sebastian Schuhmann**: Investigation, Resources, Data curation. **Roland Balbierer**: Investigation, Resources, Data curation. **Gisela Guthausen**: Supervision, Project administration, Funding acquisition. **Hermann Nirschl**: Supervision. **Gudrun Thäter**: Supervision. **Mathias J. Krause**: Supervision, Project administration, Funding acquisition, Software.

Comment
This paper forms Chapter 8, and was published with a *CRediT authorship contribution statement.*

Appendix B

Implementation

The here presented methods were implemented in the open-source software OpenLB[1]. The following table lists the presented test cases and the related commit hashes on the master branch of the OpenLB git repository[2].

Description	Usage	Commit hash	Folder name in `apps/fabiank/`
Porous media simulation	Section 2.4	864a88cfe	`porousPoiseuille3d`
Minimising the Rosenbrock function	Section 4.2.5	0fb6a3c35	`testRosenbrock`
CFD-MRI pipe flow	Chapter 6	3df7e0cab	`nmrValve3d`
Cube identification	Section 7.3	98628060c	`testDomain3d`
Stanford bunny identification	Section 7.4	957da351d	`bunny3d`
Synthetic noise reduction	Chapter 8	267d13c13	`noise3d`
Porous structure application	Chapter 9	5c9396ae2	`porousNMR3d`

[1] `https://www.openlb.net/`
[2] `https://gitlab.com/openlb/olb`