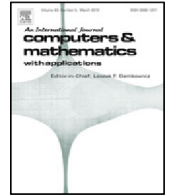




Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Solving fluid flow domain identification problems with adjoint lattice Boltzmann methods

Fabian Klemens^{a,*}, Benjamin Förster^a, Márcio Dorn^c, Gudrun Thäter^a,
Mathias J. Krause^{a,b}^a Institute for Applied and Numerical Mathematics (IANM), Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany^b Institute for Mechanical Process Engineering and Mechanics (MVM), Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany^c Institute of Informatics (INF), Federal University of Rio Grande do Sul (UFRGS), Av. Bento Gonçalves 9500, Porto Alegre, Rio Grande do Sul, Brazil

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Optimisation
Adjoint LBM
LBM
Porous media model
Fluid flow control
Domain identification

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.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Solving fluid flow domain identification problems numerically is a very important task, not only for academic studies, but also for medical and process engineering applications. One possible application is the improvement of Phase Contrast MRI (PC-MRI) measurements through coupling of measured and simulated results, which can lead to a significantly more accurate characterisation of flows and flow domains in complex geometries. The so improved PC-MRI promises to expand the area of its application and further enables saving costly and limited resources or improving the effectiveness of their use. The suggested approach has therefore an enormous economic and social impact. A great challenge thereby is to numerically solve the problem in a time reasonable for practical applications. Fluid flow domain identification problems can be formulated as restricted optimisation problems, where the difference of a measured and simulated fluid flow field is minimised. Because the side condition to be solved, here a fluid flow problem, requires a very large amount of computer resources, a highly efficient and scalable parallelisation of the solution strategy is vital.

Because of the simplicity of the core algorithm and its local computations, the lattice Boltzmann methods (LBM) have not only become a widely accepted numerical tool to solve incompressible Navier–Stokes equations [1], but also for high performance computing (HPC) and various physical problems [2,3]. However, for fluid flow control and optimisation problems its usefulness has only been partially discussed. For example for 2D topology and design optimisation by Pingen

* Corresponding author.

E-mail address: fabian.klemens@kit.edu (F. Klemens).

Nomenclature

Abbreviations

ALB	Adjoint lattice Boltzmann
ALBM	Adjoint lattice Boltzmann method
BGK	Bhatnagar, Gross and Krook
CFD	Computational fluid dynamics
HPC	High Performance Computing
LB	Lattice Boltzmann
LBM	Lattice Boltzmann method

Fluid Constants, Parameters and Variables

φ	Adjoint particle density function
$dM_{f, B\alpha}^{eq}$	Adjoint Porous Media Maxwellian distribution
α	Control
α_0	Control start value
ρ_f, ρ_{fh}	Density, discrete density
J	Goal functional
G_h	Grid term
ν	Kinematic viscosity
K	Permeability
d_h	Porosity
$M_{f, B\alpha}^{eq}$	Porous media Maxwellian distribution
f	Particle density function
B	Projection operator
Re	Reynolds number
\mathbf{G}	Side condition function
R	Universal gas constant
$\mathbf{u}, \mathbf{u}_f, \mathbf{u}_{fh}$	Velocity of flow (macroscopic, mesoscopic, discrete)
\mathbf{u}^*	Velocity of flow of the measurement
\mathbf{c}	Velocity of molecules (microscopic)

Norms and Spaces

$\mathbf{x} \cdot \mathbf{y}$	Standard scalar product for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n, n \in \mathbb{N}$
$\ \mathbf{x}\ $	Euclidean norm, $\sqrt{\mathbf{x} \cdot \mathbf{x}}$ for $\mathbf{x} \in \mathbb{R}^n, n \in \mathbb{N}$
d	Dimension of the position space $\Omega, d = 2, 3$
$\mathbb{R}_{\geq 0}$	Positive real numbers with zero, $x \in \mathbb{R}$ with $x \geq 0$
h	Discretisation/model parameter, $h \in \mathbb{R}_{>0}$
Ω	Domain filled by the fluid, $\Omega \subseteq \mathbb{R}^d$
Ω_h	Lattice, uniform mesh with spacing $h, \Omega_h \cong X_h \subseteq \mathbb{N}^d$
I	Time interval $[t_0, t_1] = I \subseteq \mathbb{R}, 0 \leq t_0 < t_1 < \infty$
I_h	Discrete time interval, $I_h := \{t \in I : t = t_0 + h^2k, k \in \mathbb{N}\}$
$\ \xi\ _{L^2(X)}$	$L^2(X)$ -norm over $X \subseteq \mathbb{R}^d, \ \xi\ _{L^2(X)} := (\int_X \xi(x) dx)^{1/2}$ for $\xi \in L^2(X)$
$\ \xi^h\ _{L^2(X_h)}$	$L^2(X_h)$ -Norm over $X_h \subseteq \mathbb{N}^d, \ \xi^h\ _{L^2(X_h)} := (\frac{1}{h^d} \sum_{x \in X_h} \xi(x))^{1/2}$ for $\xi^h \in L^2(X_h)$

Differential Operators

$\frac{d}{dx}$	Total derivative
$\frac{\partial}{\partial x}$	Partial derivative
$\nabla_{\mathbf{x}}$	Gradient
$\nabla_{\mathbf{x}} \cdot$	Divergence

et al. [4–6], which has been extended for transient flows by Kirk et al. [7]. These methods use an adjoint sensitivity approach to calculate the needed gradient based on the discrete lattice Boltzmann equation. This approach leads to a linear system, which is then solved by e.g. the Schur-complement method [5], or for transient problems by matrix–vector products due to backwards in time evaluation [7]. The first to derive an adjoint lattice Boltzmann equation were Tekitek et al. [8], whereby

the adjoints were also derived from the discrete equations. In contrast, the strategy proposed by Krause et al. [9] follows a sensitivity-based so-called *first-optimize-then-discretise* approach [10], also called *first-differentiate-then-discretise* [11], where the adjoint equation is derived on a continuous basis and then discretised afterwards. This discretisation strategy, referred to as adjoint lattice Boltzmann method (ALBM) [9], leads to algorithms as efficient as LB algorithms due to similar locality properties. Recently, Liu et al. [12] and also Laniewski-Wollk et al. [13] showed that using the first-discretise-then-optimize also yields structures similar to LBM and thus no matrix solving is required. ALBM is most often used for solving topology optimisation problems, e.g. by Liu et al. [12], Yaji et al. [14], Laniewski-Wollk et al. [13], Yonekura et al. [15] while its application to other inverse or optimisation problems is rarely studied.

The main aim of this article is to improve the robustness of the adjoint lattice Boltzmann method (ALBM) for solving fluid flow domain identification problems, proposed by Krause et al. [16]. Additionally, the method is to be thoroughly validated for laminar fluid flow regimes with existing steady state.

In order to determine the flow domain and the physical flow dynamics, it is proposed to formulate the fluid flow problem parametrised with a porous media model, where porosities are linked to control variables. The use of porosities thereby guarantees continuous functions, which are necessary for the solution strategy considered here. The optimisation problem is then minimising the difference between resulting and the measured velocity distribution. After completion of the gradient-based optimisation algorithm (e.g. the LBFGS scheme [17]), a porosity threshold value is chosen in order to interpret the resulting porosity values as solid or fluid. In order to enable solving an optimisation problem with different resolutions without adapting the initial set of control variables, i.e. to guarantee the robustness of the parametrisation, a linking is proposed which connects the variables representing a lattice-dependent porosity with control variables. Hereby, it is paid attention that a given control parameter set yield to results which are independent of the underlying grid resolution.

In Section 2 the optimisation problem is introduced, and with it a parametrised BGK–Boltzmann model for fluid flow through porous media, as well as a sensitivity-based strategy to solve fluid flow domain identification problems. In Section 3 the discretisation for the primal and dual problem is discussed, thereby the similar structure of both problems is utilised which leads to discretisation methods referred to as lattice Boltzmann method and adjoint lattice Boltzmann method. In Section 4, a robust parametrisation is introduced. It is shown that the proposed relation between the lattice-dependent porosities and the control variables are robust with respect to grid resolution. Further, the result is a projection which turns the optimisation problem in an unrestricted problem which enables the use of standard gradient-based optimisation schemes as Steepest Decent or L-BFGS.

2. Method

Here, a brief overview of the used method and the later needed variables and terms are provided. Thereby, the governing equations are first derived in continuous description and discretised afterwards, a procedure often referred to as the *first-optimize-then-discretise* strategy [18]. Further, the underlying porous media BGK–Boltzmann equation is introduced, a continuous solution strategy for the optimisation problem is given by formulating a primal and dual problem, and the specific domain identification problem equations are formulated. The method is described in detail in the preceding work of Krause et al. [16].

2.1. Overview

We consider the following constrained optimisation problem

$$\text{minimise } J(f, \alpha), \text{ such that } \mathbf{G}(f, \alpha) = 0. \quad (1)$$

Thereby f is said to be the *state*, α the *control*, J the *goal function* and $\mathbf{G}(f, \alpha) = 0$ the *side condition* or *constraint*. Using a quasi-Newton method, like BFGS [17], the optimisation problem is solved iteratively using only the goal function J and its total derivative with respect to the control $\frac{d}{d\alpha}J$ (cf. [9,16]). However, finding the total derivative or calculating it efficiently is the main challenge for optimisation problems.

2.2. Side condition

Here, the side condition is an element of the *diffusive limit family of porous media BGK–Boltzmann equations* [19,16] denoted as

$$h^2 \frac{d}{dt}f + \frac{1}{3\nu} \left(f - M_{f,d_h}^{eq} \right) = 0, \quad (2)$$

where $f = f(t, \mathbf{r}, \mathbf{c})$ is the *particle distribution function*, with time $t \in I = [t_0, t_1] \subseteq \mathbb{R}_{\geq 0}$, position $\mathbf{r} \in \Omega \subseteq \mathbb{R}^d$, velocity $\mathbf{c} \in \mathbb{R}^d$, model parameter $h \in \mathbb{R}_{>0}$ and kinematic viscosity $\nu \in \mathbb{R}$. As shown by Saint-Raymond [20] Eq. (2) is related to an incompressible Navier–Stokes equation. The macroscopic *density* and *velocity* ρ_f, \mathbf{u}_f of the Newtonian fluid can be obtained as moments of f via

$$\rho_f := \int_{\mathbb{R}^d} f(\mathbf{v}) d\mathbf{v} \quad \text{and} \quad \mathbf{u}_f := \frac{1}{\rho_f} \int_{\Omega} \mathbf{v} f(\mathbf{v}) d\mathbf{v}.$$

Furthermore,

$$M_{f,d_h}^{eq} = \frac{\rho_f h^d}{\left(\frac{2}{3}\pi\right)^{d/2}} \exp\left(-\frac{3}{2}(\mathbf{c} h - d\mathbf{u}_f h)^2\right) \quad \text{in } I \times \Omega \times \mathbb{R}^d$$

denotes the *Porous Media Maxwellian distribution*, where the *porosity* $d : \Omega \rightarrow [0, 1]$ is defined by (c.f. [21,16])

$$d(r) = 1 - \frac{\nu\tau}{K} \tag{3}$$

with *permeability* K . Porosity values of $d := d(r)$ are to be interpreted as solid ($d = 0$), fluid ($d = 1$) and porous ($d \in (0, 1)$) at point $r \in \Omega$.

Using the porous media model requires the control parameter $\alpha(r) \in \mathbb{R}$ to be projected onto the porosity parameter $d_h(r) \in [0, 1]$ for all points $r \in \Omega$ through an operator $B\alpha = d_h$. An appropriate operator B is introduced in Section 4. This coupling of porosity and control leads to

$$\mathbf{G}(f(\alpha), \alpha) = h^2 \frac{d}{dt} f + \frac{1}{3\nu} (f - M_{f,B\alpha}^{eq}) . \tag{4}$$

2.3. Goal function

For domain identification problems the goal functional J is defined as

$$J(f, \alpha) = \frac{1}{2} \int_{\Omega_j} (\mathbf{u}_f - \mathbf{u}^*)^2 dr , \tag{5}$$

with \mathbf{u}^* is the *measured flow field* (e.g. by a PC-MRI scan) and $\Omega_j \subseteq \Omega$ is the *objective domain*. The partial derivative with respect to the particle distribution, needed for the *optimality condition* (Section 2.4), reads

$$\frac{\partial}{\partial f} J(f, \alpha) = -\frac{(\mathbf{u}^* - \mathbf{u}_f)(\mathbf{v} - \mathbf{u}^*)}{\rho_f} .$$

Note that the evaluation of the goal functional J requires solving the side condition $\mathbf{G}(f, \alpha) = \mathbf{0}$, which corresponds to solving a fluid flow problem.

2.4. Optimality condition

A necessary condition for an optimal state (f^*, α^*) is given by the *optimality condition* $\frac{d}{d\alpha} J(f^*, \alpha^*) = 0$ [9,16]. Here, the optimality condition formulates as

$$\frac{d}{d\alpha} J(f(\alpha), \alpha) = u\tau \int_{\mathbb{R}^d} \varphi 3h^2(v - B\alpha\mathbf{u}_f)M_{f,B\alpha}^{eq} dv + \frac{\partial}{\partial f} J(f, \alpha) , \tag{6}$$

where φ is determined by solving the *adjoint porous media BGK–Boltzmann equation* (cf. [16])

$$\frac{d}{dt} \varphi - \frac{1}{3\nu}(\varphi - dM_{f,B\alpha}^{eq}) + \frac{\partial}{\partial f} J = 0 , \tag{7}$$

with the *adjoint Maxwellian distribution*

$$dM_{f,B\alpha}^{eq} = \int_{\mathbb{R}^d} \varphi(\hat{\mathbf{v}}) \frac{(3h^2 (\mathbf{u}_f - \mathbf{v}) (B\alpha\mathbf{u}_f - \hat{\mathbf{v}}) B\alpha + 1)}{\rho} M_{f,B\alpha}^{eq}(\hat{\mathbf{v}}) d\hat{\mathbf{v}} .$$

The main advantage of this method is the similarity of the porous media BGK–Boltzmann equation (2) and its adjoint formulation (7), as they both can be discretised in a similar manner, which is discussed in Section 3. The discretised form of the optimality condition (6) is then used for a gradient-based method like steepest descent or BFGS in combination with e.g. the Armijo or the Wolfe–Powell rule to find the optimal state iteratively (cf. [22,16]).

3. Discretisation

Due to the similar structure of the LB equation (2) and the ALB equation (7), similar discretisation methods can be used, which are referred to as lattice Boltzmann method (LBM) and adjoint lattice Boltzmann method (ALBM), depending on the based equations.

LBM are strategies for discretising a family of Boltzmann equations or of other mesoscopic Boltzmann-like equations which is related to a target equation in a certain limit which is usually macroscopic in nature. The basic and common idea of all LBM is the coupling of discretisation parameters with those parameters characterising the limit process.

The specific domain identification problem equations are discretised with an adjoint lattice Boltzmann method (ALBM) [9,16].

3.1. Lattice Boltzmann method

The continuous space $I \times \Omega \times \mathbb{R}^d$ is replaced by a discrete space $I_h \times \Omega_h \times Q$ where h is identified with the model parameter and is now called the discretisation parameter. The position space Ω_h is chosen as a uniform grid with spacing h and the discrete time interval is set to $I_h := \{t \in I : t = t_0 + kh^2, k \in \mathbb{N}\}$. The velocity space Q consists of $q \in \mathbb{N}$ directions \mathbf{c}_i ($i = 0, 1, \dots, q - 1$) which link dedicated neighbouring positions in such a way that for $\mathbf{r} \in \text{int } \Omega_h$ it holds $\mathbf{r} + \mathbf{c}_i h^2 \in \Omega_h$, i.e. $\mathbf{c}_i \sim h^{-1}$. The resulting discrete phase space is called the *lattice* and denoted by $DdQq$. The continuous distribution function f is replaced by a set f^h of q distribution functions f_i ($i = 0, 1, \dots, q - 1$), which represent an average value of f in the neighbourhood of the velocity \mathbf{c}_i . For the porous media model the lattice porosity d_h is defined as [16,21,23]

$$d_h = 1 - h^2 \frac{v_h \tau_h}{K}, \tag{8}$$

with physical permeability K and relaxation time $\tau_h = 3v_h + 1/2$. Where the parameter h^2 comes from non-dimensionalisation of the physical permeability. The iterative process in an LB algorithm can be written in two steps as follows, the *collision step* (9) and the *streaming step* (10):

$$\tilde{f}_i(t, \mathbf{r}) = f_i(t, \mathbf{r}) - \frac{1}{3v_h + 1/2} \left(f_i(t, \mathbf{r}) - M_{f_i, d_h}^{eq}(t, \mathbf{r}) \right), \tag{9}$$

$$f_i(t + h^2, \mathbf{r} + \mathbf{c}_i h^2) = \tilde{f}_i(t, \mathbf{r}) \tag{10}$$

for $i = 0, 1, \dots, q - 1$, where

$$M_{f_i, d_h}^{eq}(t, \mathbf{r}) := \frac{w_i}{w} \rho_{fh} \left(1 + 3h^2 \mathbf{c}_i \cdot d_h \mathbf{u}_{fh} - \frac{3}{2} h^2 (d_h \mathbf{u}_{fh})^2 + \frac{9}{2} h^4 (\mathbf{c}_i \cdot d_h \mathbf{u}_{fh})^2 \right) \tag{11}$$

is a *discretised Porous-Media Maxwell distribution* with moments ρ_{fh} and \mathbf{u}_{fh} which are defined as

$$\rho_{fh} := \sum_{i=0}^{q-1} f_i \quad \text{and} \quad \mathbf{u}_{fh} := \frac{1}{\rho_{fh}} \sum_{i=0}^{q-1} \mathbf{c}_i f_i,$$

where \mathbf{u}_{fh} corresponds to the *macroscopic fluid velocity* and ρ_{fh} to the *mass density*. The lattice viscosity v_h is assumed to be given, and the terms $w_i/w, \mathbf{c}_i h$ ($i = 0, 1, \dots, q - 1$) are model dependent constants. An exhaustive derivation of various LB equations can be found e.g. in [1,24,25].

For $D2Q9$ and $D3Q27$ Krause [26] showed that the truncation error comparing an element of the diffusive limit family of BGK–Boltzmann equations with its corresponding discrete LB term is of second order. Krause’s approach is different to previously published derivations of LBM, where macroscopically motivated assumptions are made. For the derivation of the ALBM the approach in [26] is followed.

3.2. Adjoint lattice Boltzmann method

Like for the LBM (cf. [1,24,27]), the particular choice of $I_h \times \Omega_h \times Q$ sets up an ALBM model which is denoted by $DdQq$ with d representing the dimension and q the number of discrete velocities in $Q \subseteq \mathbb{R}^d$. Commonly $D2Q9, D3Q19$ and $D3Q27$ models are applied. In a similar manner an iterative algorithm can be derived, which is executed step by step but for decreasing $t \in I_h$. In every time step two operations are to be performed for all $\mathbf{r} \in \Omega_h$ and every $j = 0, 1, \dots, q - 1$, namely the *adjoint collision step* (12) and the *adjoint streaming step* (13)

$$\tilde{\varphi}_j(t, \mathbf{r}) = \varphi_j(t, \mathbf{r}) - \frac{1}{3v_h + 1/2} \left(\varphi_j(t, \mathbf{r}) - dM_{f^h, B\alpha}^{eq}(t, \mathbf{r}) \right) + \frac{6v_h}{6v_h + 1} h^2 dJ_{f^h, B\alpha}(t) \tag{12}$$

$$\varphi_j(t - h^2, \mathbf{r} - h^2 \mathbf{c}_j) = \tilde{\varphi}_j(t, \mathbf{r}), \tag{13}$$

where $\varphi_j(t, \mathbf{r}) := \varphi(t, \mathbf{r}, \mathbf{c}_j)$ and $\mathbf{c}_j \in Q$, and with *velocity discrete adjoint Maxwellian distribution*

$$dM_{f^h, B\alpha}^{eq}(\mathbf{c}_j) := \sum_{i=0}^{q-1} \varphi_i(\mathbf{c}_i) \frac{(3B\alpha (B\alpha h \mathbf{u}_{fh} - \tilde{\mathbf{c}}_j) (h \mathbf{u}_{fh} - \tilde{\mathbf{c}}_i) + 1)}{\rho_{fh}} M_{f^h, B\alpha}^{eq}.$$

The *velocity discrete derivative of the goal functional* reads

$$dJ_{f^h, B\alpha}(\mathbf{c}_j) = - \frac{(\mathbf{u}^* - \mathbf{u}_{fh})(\mathbf{c}_j - \mathbf{u}^*)}{\rho_{fh}}.$$

The advantage of the ALBM is that the resulting equations (cf. (12), (13)) are very similar to the LB equations (cf. (9), (10)). The main differences are its time reverse character and the additional term $\frac{6v_h}{6v_h+1} h^2 dJ_{f^h}$. However, its locality properties basically remain the same.

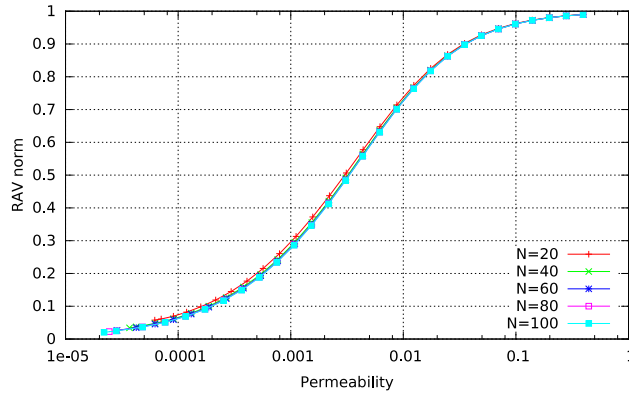


Fig. 1. Relative Average Velocity Norm (RAV Norm) over Permeability K . We observe that the RAV norm is approximately the same for all grid resolutions N . This is a strong indication for the grid-independence of permeability K in a sense that for different grid sizes, the same value of permeability leads to the same fluid flow behaviour.

Consequently an ALB algorithm can be implemented similarly to a LB algorithm [16]. In particular, the data structure design and the hybrid parallelisation strategy proposed early for standard LBM in [28] can be applied. Therefore, executing an ALB scheme is expected to qualitatively perform as efficient as a LB scheme. For a detailed analysis and efficient implementation the preceding work of Krause et al. [16] is recommended.

4. Projection: linking porosity and control

The proposed optimisation problem is unbounded, i.e. all control variable are $\alpha \in \mathbb{R}$, but to change the topology of our domain we need to control the porosity parameter d_h which is in $[0, 1]$. This section is dedicated to finding a projection $B : \mathbb{R} \rightarrow [0, 1]$ with $\alpha \rightarrow B\alpha := d_h$, linking each optimisation parameter $\alpha \in \mathbb{R}$ to a porosity value $d_h \in [0, 1]$. Thereby the proposed projection should preserve the physical properties of fluid flow in porous media, such that the optimisation problem can be solved for different resolutions without adapting the initial set of control variables.

4.1. Analysis of lattice porosity and permeability in porous media LBM

In order to analyse the influence of the lattice porosity on the simulation, we define a simple test case and a measure of the average flow velocity through a certain domain within a flow field. The test case is a cube of side length 1 m extending from geometrical point $(0, 0, 0)$ to $(1 \text{ m}, 1 \text{ m}, 1 \text{ m})$, defined as the flow domain Ω_h with constant inflow at the surface of the cube at $y = 0$ and constant outflow at $y = 1$. At the inflow and outflow plane the velocity is set to 1 m/s in y -direction (cf. Fig. 4), with viscosity of $0.1 \text{ m}^2/\text{s}$. Boundary conditions at $x = 0, x = 1, z = 0$ and $z = 1$ are set to bounce back. The design domain D (green), where a fixed porosity d_h is set, is defined as a cube of side length 0.4 m in the centre of Ω , extending from $(0.3, 0.3, 0.3)$ to $(0.7, 0.7, 0.7)$ (cf. Fig. 4). This test case is also used in the numerical experiments in Section 5.

For a fixed porosity value d_h , we define the L^2 norm of the velocity inside the cube D

$$u^{d_h} := \|u\|_{L^2(D)} = \left(\int_D u^2 dr \right)^{1/2}, \tag{14}$$

where $d_h(r) = d_h$ for all $r \in D$ and u being the flow velocity function. We define the relative average velocity (RAV) norm $u_{rel}^{d_h}$ as the relative change of the velocity field inside the design domain for a certain porosity d_h with respect to the velocity field of full fluid flow ($d_h = 1$)

$$u_{rel}^{d_h} := \frac{u^{d_h}}{u^1}. \tag{15}$$

The simulation was performed for five different grid sizes $h = 1/N$ and twenty-eight permeability values K which are translated into different porosity values d_h by relation $d_h = 1 - h^2 \nu_h \tau_h / K$ (cf. (8)). The results are shown in Fig. 1, which indicate that the physical permeability is grid-independent in the sense that it does not change the fluid flow behaviour for higher resolutions. The presented result is expected as it only indicates the validity of the used porous media model [21].

The problem which arises is the grid dependency of porosity due to Eq. (8), therefore resulting in grid dependency of the optimisation parameter if used directly, for example in optimisation methods with bounded sets. The relation of porosity and permeability is shown in Fig. 2 for a viscosity of $u_h = 0.002$. It shows that for a given permeability the corresponding porosity changes with the resolution. In order to maintain the fluid behaviour for different resolutions, we must therefore take permeability into account.

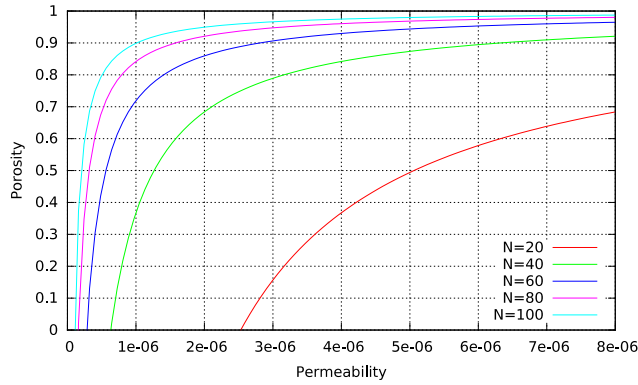


Fig. 2. Porosity d_h over Permeability K for different grid resolutions N . This is the graph of relation (8) for $u_h = 0.002$. Here the grid dependency of the relation porosity and permeability can be seen. This is in contrast to Fig. 1, where the grid-independency of the permeability is indicated.

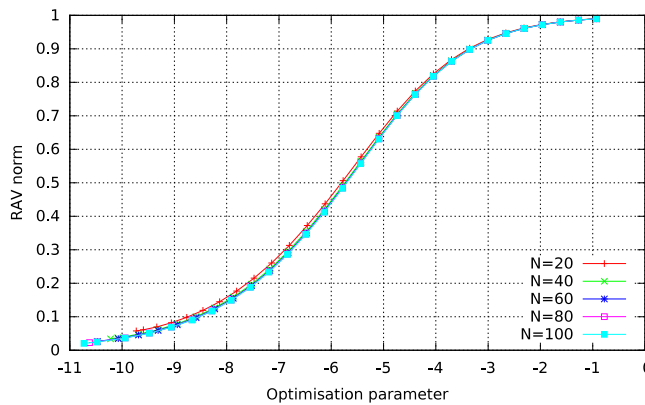


Fig. 3. RAV norm of projection B for multiple grid resolutions N . Using the proposed projection it can be seen that for different resolutions the fluid behaviour is very similar for fixed optimisation parameters. This is a good indication of the grid-independency of the proposed projection.

4.2. Constructing a projection using permeability

To combine the advantages of grid-independency of permeability with the usability of porosity, we use the relation of the porous media model (cf. (8)). After defining the *grid term* $G_h := h^2 \nu_h \tau_h$ for notation purpose, we get

$$d_h = 1 - \frac{G_h}{K}, \tag{16}$$

where $K \in [G_h, \infty)$. We want the control to have the same properties as the permeability, therefore we are looking for a subprojection $p : \mathbb{R} \rightarrow (0, \infty)$ such that $p(\alpha) + G_h \in [G_h, \infty)$, which leads to

$$B\alpha = 1 - \frac{G_h}{p(\alpha) + G_h}. \tag{17}$$

We propose $p(\alpha) := e^\alpha$ which yields

$$B\alpha = 1 - \frac{G_h}{e^\alpha + G_h} = \frac{e^\alpha}{e^\alpha + G_h}, \tag{18}$$

with $B \in C^\infty(\mathbb{R})$ and $B\alpha \in (0, 1)$. Analysing the projection with the RAV norm, a similar fluid behaviour for a given control is observed, see Fig. 3. This is expected as Fig. 1 shows the logarithmic plot of K , where now the control is mapped to $\alpha = \log(K - G_h)$ with $G_h \xrightarrow{h \rightarrow 0} 0$. Note, that using $\alpha = \log(K)$ directly would lead to $d_h \in (-\infty, 1)$ and could therefore not be used in the porous media model.

To summaries, the proposed projection $B\alpha = \frac{e^\alpha}{e^\alpha + G_h}$ is a smooth function, which maps the control $\alpha \in \mathbb{R}$ to a porosity $d_h \in (0, 1)$. Thereby the control is linked to the physical permeability which is consistent with the used porous media model.

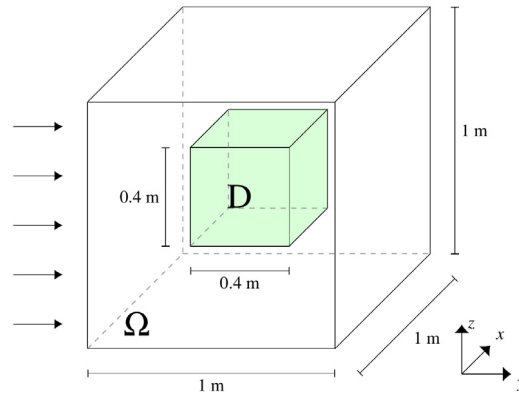


Fig. 4. Object Identification Setup. A cube of side length 1 m extending from geometrical point $(0, 0, 0)$ to $(1, 1, 1)$ is defined as the flow domain Ω with constant inflow at $y = 0$ and constant outflow at $y = 1$. Boundary conditions at $x = 0, x = 1, z = 0$ and $z = 1$ are set to bounce back. The design domain D (green) is defined as a cube of side length 0.4 m in the centre of Ω , extending from $(0.3, 0.3, 0.3)$ to $(0.7, 0.7, 0.7)$. Different object geometries C are placed inside the design domain which are then attempted to be identified by the optimisation algorithm.

5. Numerical experiments

In this section, different flow domain identification test cases are investigated for validation. First, the novel domain identification approach of *a priori* defined geometries is presented in detail. Then, a simple test case of identifying a cube in the centre of a flow domain is presented, with the objective domain subsequently being reduced to subsets of the flow data.

5.1. Test case and generic evaluation approach

For validation of the flow domain identification method, a generic test case approach is set up, where the geometry information which is to be identified is available *a priori*. Hereby, first the fluid flow around a solid object geometry Π is simulated and the resulting flow field is saved. Then the optimisation algorithm is applied to the obtained flow field by choosing a start value α_0 and setting $d_h(r) = B\alpha_0$ for all points r in the design domain $D \supseteq \Pi$ (see Fig. 4). Algorithm 1 describes the generic approach of *a priori* defined geometries in more detail.

1. Simulate fluid flow with $d_h(r) = 1$ for $r \notin \Pi$ and $d_h(r) = 0$ for $r \in \Pi$;
2. Save the resulting flow data as the solution flow field u^* ;
3. Choose start value α_0 and set $d_h(r) = 1$ for $r \in \Omega \setminus D$ and $d_h(r) = B\alpha_0$ for all points $r \in D$;
4. Solve optimisation problem with the ALBM-based method presented in Section 2;

Algorithm 1: Generic validation approach with an *a priori* defined geometry Π .

The quality of the resulting object geometry is evaluated by comparison to the *a priori* defined geometry Π .

The modelling and discretisation of the fluid flow simulation follows the approach of *collide and stream* presented in Section 3. The modelling and discretisation of the optimisation problem follows the approach of ALBM described in Sections 2 and 3. The simulations and optimisations are run on a regular grid of resolution $N = 100$, i.e. a lattice length of $1/N = 0.01$, if not noted otherwise. In all test cases, the gradient-based quasi-Newton algorithm BFGS [17] is used. For the physical properties of the object identification test case a viscosity of $\nu = 0.1 \text{ m}^2/\text{s}$ is used which results in a Reynolds number of 10.

The geometrical setup of these test cases consists of a cube of side length 1 m, which is defined as the flow domain Ω as described in Fig. 4. The design domain D (green) is defined as a cube of side length 0.4 m in the centre of Ω . Various different object geometries Π are placed inside the design domain D , which are to be identified by the solution approach.

5.2. Partial data objectives

It is a common problem with flow domain identification to have only incomplete data available. For example, with PC-MRI measurements, most of the time only slices of data are measured. This poses a big challenge to the optimisation algorithm. One may be interested in a specific object which is unfortunately not covered by the area where the measurement is applied.

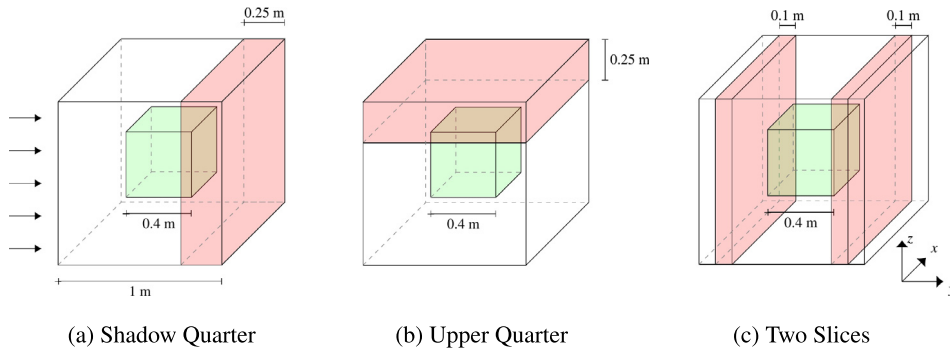


Fig. 5. Cube Identification from Partial Data (Scheme). Inside the design domain D (green) an object $C := \Pi$ is placed, which is to be identified. The objective domain (red) is reduced to different parts of the domain, and therefore only partial information of the flow data is used for the optimisation algorithm. For (a) (Shadow Quarter) the objective domain is reduced to the quarter behind the cube with respect to the flow direction, reaching from $y = 0.75$ to $y = 1$. For (b) (Upper Quarter) the upper quarter of the flow domain, reaching from $z = 0.75$ to $z = 1$ is used. And for (c) (Two Slices) one slice in front of the object, reaching from $y = 0.1$ to $y = 0.2$, and one slice behind the object, reaching from $y = 0.8$ to $y = 0.9$ is used. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

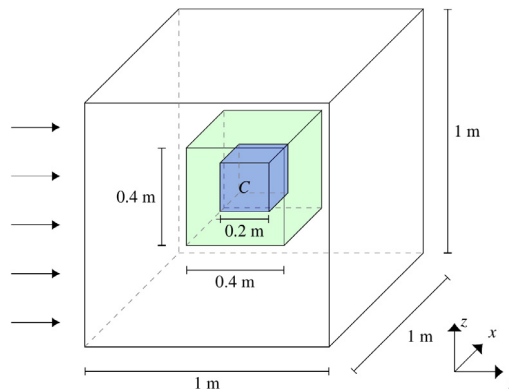


Fig. 6. Object Identification: Cube (Scheme). A cube $C := \Pi$ of side length 0.2 m (blue) is placed in the centre of the design domain D (green).

Therefore, the same cube identification test case as presented above is set up with $\Pi := C$, but the objective domain, where the objective functional is evaluated, is reduced to three different subdomains of Ω . In all three partial data test cases, the design domain D is completely uncovered by the objective functional. The partial data test cases are the *flow shadow quarter* test case where only the 25% of the flow data behind the cube is being used. An *upper quarter* test case with only the upper 25% of the flow data is being used. And the *two slices* test case where only two slices of each 10% of the flow data behind and in front of the cube is being used (cf. Fig. 5).

5.3. Test case 1: Cube identification

First a simple flow domain identification test case is set up. A single cube $C := \Pi$ of 0.2 m is placed in the centre of the previously introduced design domain D . The detailed setup of this test case is presented in Fig. 6.

The graph of the optimisation objective for the numerical results of this test case are presented in Fig. 7 for different grid resolutions $N = 25, 50$ and 100 , i.e. the voxel length is $1/N$ m. It is observed that the value of the objective decreases for increasing grid resolutions, indicating the convergence of the objective towards zero if N tends to infinity, as well as the grid independence of the start value.

In Fig. 8, a graphical visualisation of the numerical results of this test case with a porosity threshold value of $d_h = 0.99$ is presented. The results show very few points of a porosity value near the threshold value of $d_h = 0.99$ after termination of the optimisation algorithm. The cube can be clearly identified after as few as 10 optimisation steps, with a highly precise identification after 35 steps with very low porosity values, indicating a very clear separation between solid and fluid points. In Fig. 9 the porosity in a slice of the cube in the $y - z$ plane is shown. As it can be seen, the inside of the cube is also getting more solid. In order to measure the error of the object identification we propose following error norm

$$\frac{\|d_h - d_c\|_{L^2(D)}}{\|d_c\|_{L^2(D)}}, \tag{19}$$

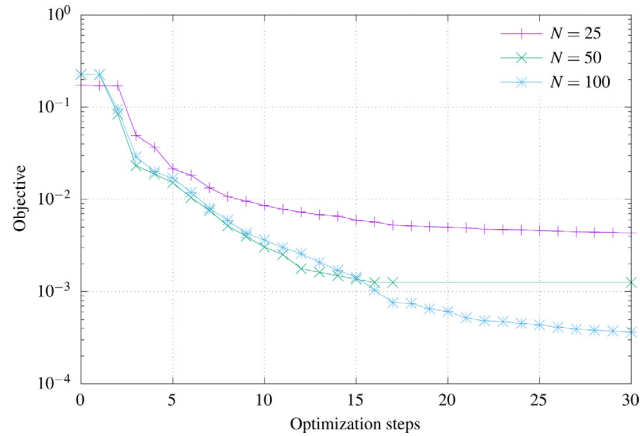


Fig. 7. Optimisation objective on different grid resolutions N . Results for the fluid flow characterisation. The optimisation algorithm reached a termination condition for grid resolution $N = 50$ after 17 optimisation steps. For $N = 25$ and $N = 100$, the number of optimisation steps exceeded 30. An artificial data point is inserted for $N = 50$ at step 30 with the same value as at step 17 for comparison purposes.

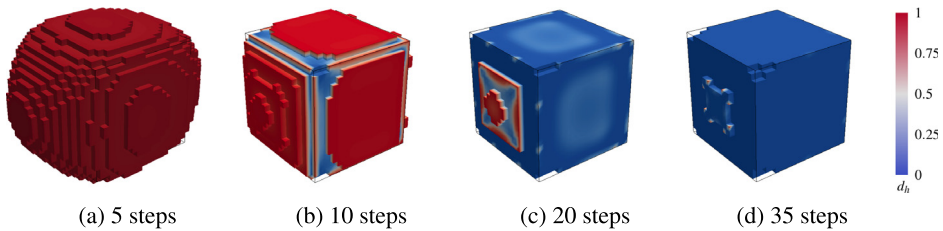


Fig. 8. Cube Identification (Results). The cube identification algorithm after 5, 10, 20 and 35 optimisation steps. The original cube is marked by a black outline. The figure shows the same section, displaying only lattice-porosities of $d_h \leq 0.99$.

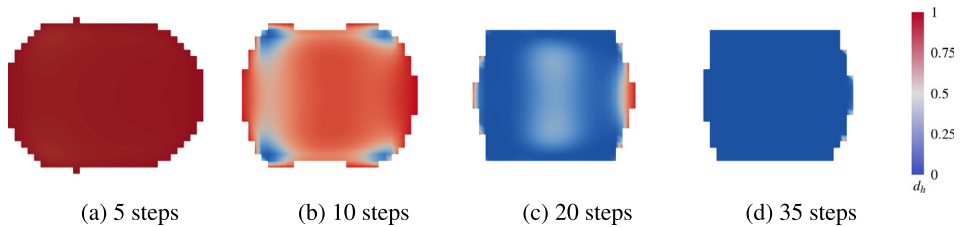


Fig. 9. Cube Identification Inside (Results). The cube identification algorithm after 5, 10, 20 and 35 optimisation steps. Shown is the porosity in a slice of the cube in the $y - z$ plane, with threshold of $d_h \leq 0.99$.

where d_h is the lattice porosity obtained by the optimisation method and d_c the porosity of the object. Fig. 10 shows the proposed error norm for the identification of the cube, where the error starts at 97% and reaches 2% in the end, indicating the clear identification of the object.

5.3.1. Test case 2: Cube identification from partial data

For the fluid flow characterisation the relative L^2 error of the velocity inside the objective domain

$$\frac{\|u - u^*\|_{L^2(\Omega)}}{\|u^*\|_{L^2(\Omega)}} \tag{20}$$

was considered. The results of the three partial data test cases and the full data test case is presented in Fig. 11. It is observed that for all three partial data test cases the objective reaches very low values, indicating that the optimisation works very well inside the objective domain. After 10 steps the error reaches near its minimum for all cases. Fig. 12 shows the error inside the whole domain. For all test cases the error gets minimised, where the full information case works best, as expected. The shadow quarter test case is the slowest of the three to converge, but still reaches an error of 5%. The two slices test case produces the best approximation in the lowest number of optimisation steps with only 20% of the flow data being used and

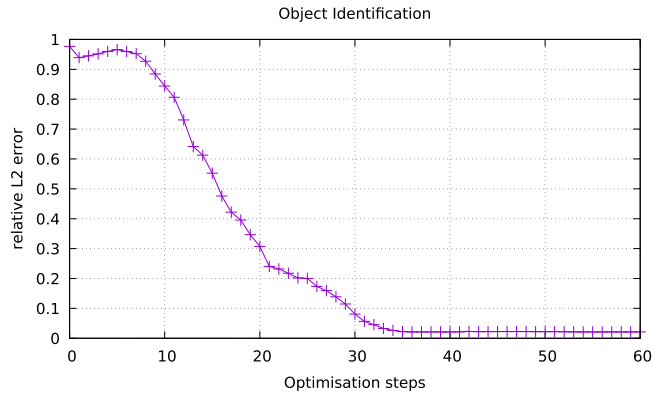


Fig. 10. Object Identification. The relative L^2 error for the given and simulated porosity fields is shown. The error starts at 97% and reaches 2%, indicating a clear identification of the object.

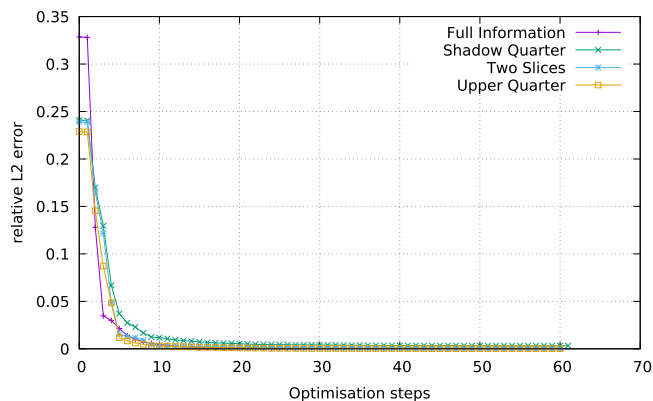


Fig. 11. Flow Characterisation: Objective Domain. The relative L^2 error inside the objective domain for the velocity is given for the full and partial information test cases.

an error of 1%. Although the fluid flow is very accurately characterised by the proposed method, the object is not as clearly identified. As can be seen in Fig. 14 a cubical form is nearly identified but with high porosities and low porosities are only found near the objective domain. This is best seen for the upper quarter case. For the two slices case the lowest porosities are in direction of the flow. Fig. 13 shows the relative L^2 error of the porosities were an error of 64% for the shadow case, 33% for the two slices case and 79% for the upper case are found. This contrasts strongly with the results obtained for the flow characterisation, where the error is less than 5%. This could be due to the interpretation of the lattice porosities, which is strongly dependent on the grid size, as discussed in Section 4. The interpretation of a lattice porosity for one grid size is a very different one for another grid size (cf. 2).

5.4. Test case 3: Identification of more complex object geometries

After successfully identifying a simple cube in the design domain D in the previous test cases, the optimisation algorithm is applied to increasingly complex geometries. For each of the following test cases, a scheme is provided, describing the *a priori* defined geometry Π that is to be identified by the optimisation algorithm and the numerical results are visualised for validation.

5.4.1. Object identification: Sphere

In this test case, a sphere $S = \Pi$ of radius 0.1 m (blue) is placed in the centre of the design domain (green) as described in Fig. 15.

The numerical results are visualised in Fig. 16, the result of the relative L^2 error of the objective is shown in Fig. 17. It is observed that the optimisation algorithm is able to identify the position and shape of the sphere at a satisfying level after 5 steps and with very low porosity values after only 15 steps.

After 2 steps, the object is barely to be recognised, but after 5 steps, the sphere is mostly identified. After 10 steps, the porosity values have become lower and the shape of the sphere is identified more precisely. After 15 steps, the sphere is clearly identified with very low porosity values.

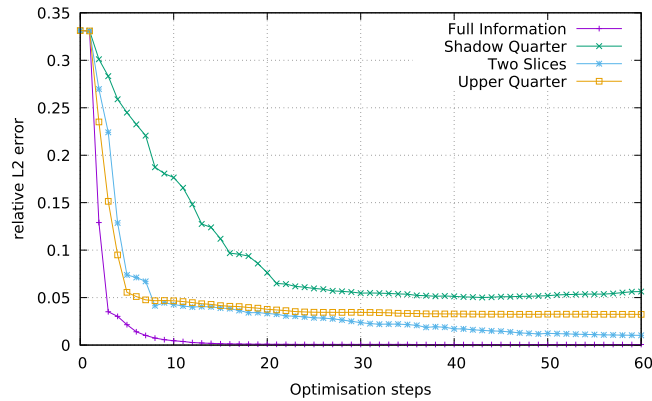


Fig. 12. Flow Characterisation: Whole Domain. The relative L^2 error for the velocity inside the whole domain is given for the full and partial information test cases. The full information case reaches the lowest error. For the partial information cases the shadow case shows the highest error at 5%, where the two slices case reaches 1%.

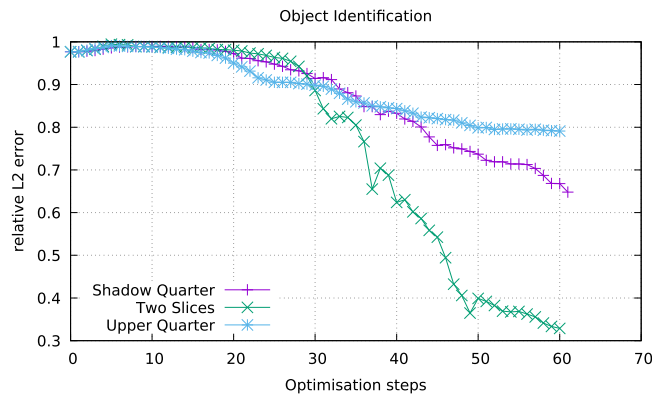


Fig. 13. Object Identification. Shown is the relative error of given and simulated porosity fields for the partial test cases.

5.4.2. Two spheres side by side

In this test case, two spheres $\{S_1, S_2\} =: \mathcal{I}$ are placed side by side in the centre of the design domain as described in Fig. 18. It is assumed that the area in the middle of the two spheres poses a big challenge for the optimisation algorithm.

The visualisation of the numerical results are presented in Fig. 19, the result of the relative L^2 error in Fig. 20. It is observed that the optimisation algorithm is able to clearly identify the position and shape of the two spheres with very low porosity values after 25 steps.

After 5 steps, the shape of the objects is roughly to be recognised. After 15 steps, the two spheres are identified with low porosity values, whilst the front area of spheres consists still of very high porosity values. After 25 steps, the two spheres are clearly identified with very low porosity values.

5.4.3. Cube and cuboid

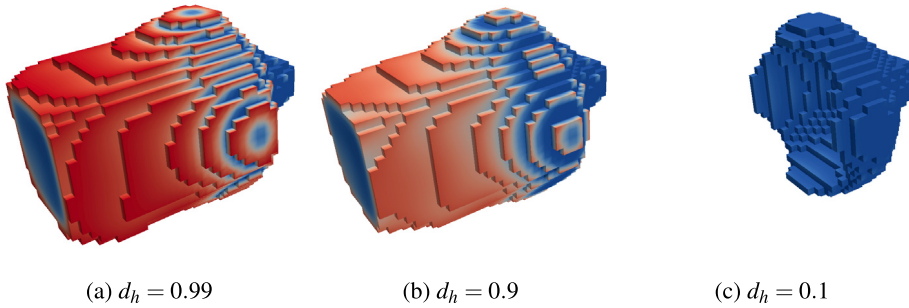
In this test case, a cube and a cuboid are placed in the design domain asymmetrically as described in Fig. 21. It is assumed that the asymmetric positioning is a challenging task for the optimisation algorithm. The close positioning of the objects is suspected to be an additional challenge.

One needs to be aware of the position of the small cube in the corner of the design domain. Since all neighbouring points outside the design domain are set to be totally fluid ($d_h = 1$), the three outwards oriented surfaces of the cube are assumed to be immediately identified as even surfaces.

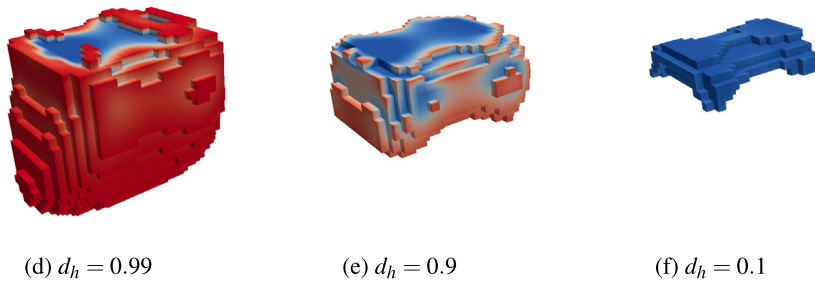
The visualisation of the numerical results are presented in Fig. 22, with the relative L^2 error of the objective in Fig. 23. It is observed that the optimisation algorithm is able to identify both objects with satisfying precision in 10 steps and with significantly lower porosity values in 25 steps.

After 2 steps, the two objects are still indistinguishable. After 5 steps, one can already roughly identify the two distinct objects. After 10 steps, the objects are mostly identified, leaving it for the next 15 steps to lower the porosity values of the points even more. After 25 steps, the objects are clearly identified.

Partial Data (Shadow Quarter)



Partial Data (Upper Quarter)



Partial Data (Two Slices)

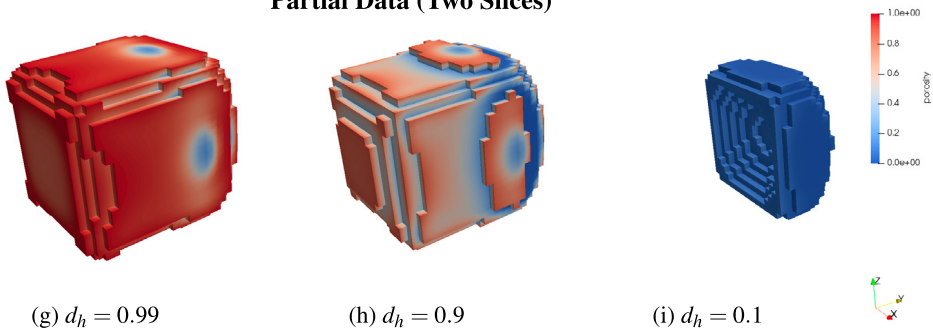


Fig. 14. Cube Identification from Partial Data (Results). Each row shows the results of the optimisation algorithm for a different partial data test case at the last optimisation step (cf. Fig. 5). The threshold for every result is given, where $d_h = 0.9$ means all porosities greater than 0.9 are not shown.

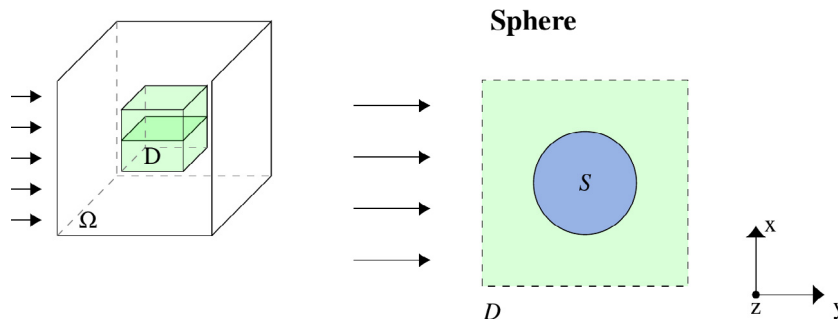


Fig. 15. Object Identification: Sphere (Scheme). A sphere $S =: \Pi$ of radius 0.1 m (blue), which is to be identified, is placed in the centre of the design domain D (green).

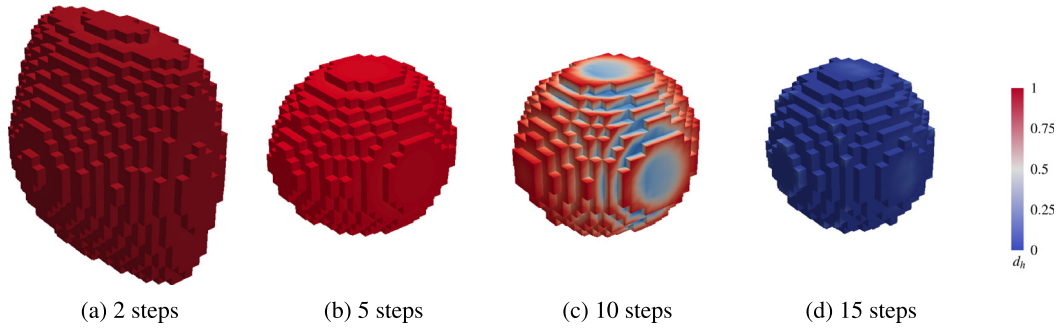


Fig. 16. Object Identification: Sphere (Results). Optimisation results after 2, 5, 10 and 15 steps are visualised for the test case described in Fig. 15.

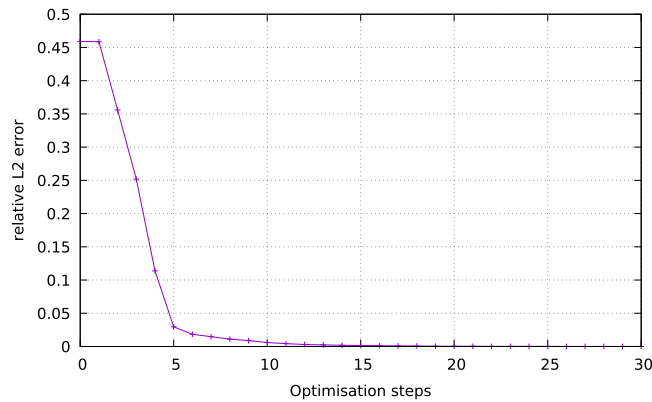


Fig. 17. Identification of a Sphere. Relative L2 error of the velocity for the full data test case of a sphere.

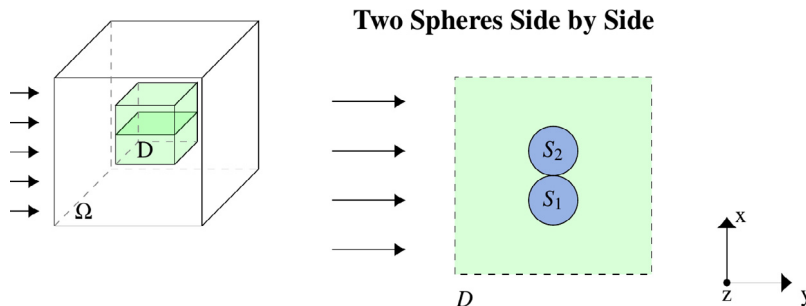


Fig. 18. Object Identification: Two Spheres Side By Side (Scheme). Two spheres $\{S_1, S_2\} =: \Pi$ with radius 0.05 m (blue), which are to be identified, are placed side-by-side in the centre of the design domain D (green). The centre of S_1 is at $(0.45, 0.5, 0.5)$ and the centre of S_2 is at $(0.55, 0.5, 0.5)$.

It is observed that the outwards oriented surfaces of the small cube are identified as even surfaces as early as in the second step.

6. Conclusion

In this work, the solution strategy for domain identification problems [16] has been improved and thoroughly validated. The approach allows the identification of objects in a fluid by solving an optimisation problem based on ALBM. Therefore, the parameters of a porous media fluid flow problem are controlled to achieve a specified velocity profile. It turns out that linking the variables representing the porosity of the discrete problem with the control variables is the key to a robust setup that does not require initial parameter tuning. The linking is analysed and a projection is proposed to ensure grid independence for a given set of control variables. Based on a porous media model formulated on a mesoscopic scale, it is shown that the novel ALBM-based optimisation approach is able to clearly identify the flow domain for problems up to 70000 controlled variables, in wide range of validation test cases, various *a priori* defined object geometries of increasing complexity and

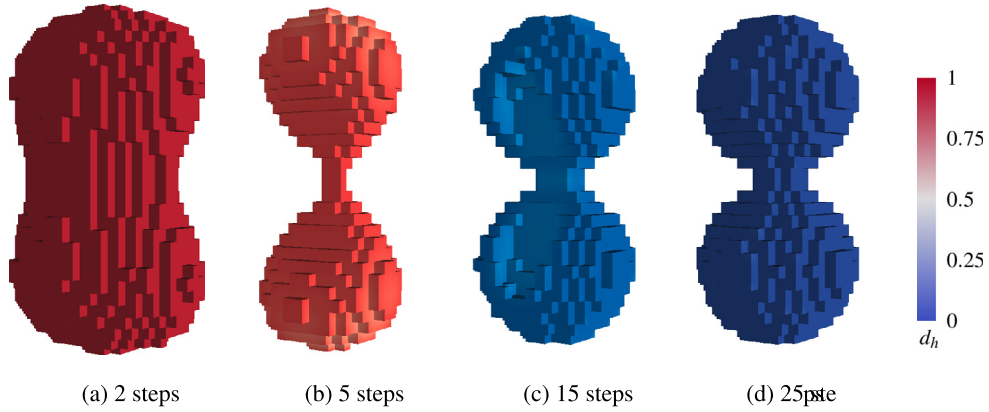


Fig. 19. Object Identification: Two Spheres Side By Side (Results). Optimisation results after 2, 5, 15 and 25 steps are visualised for the test case described in Fig. 18.

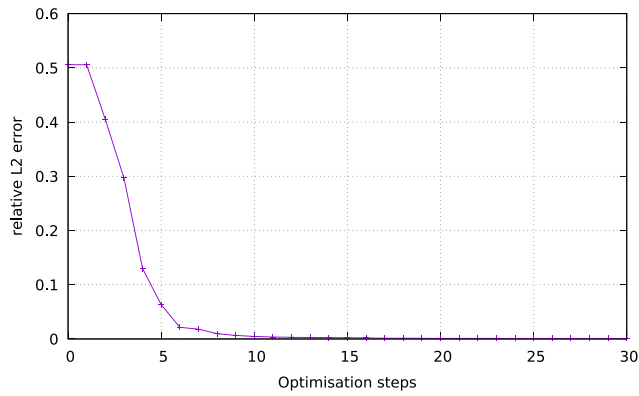


Fig. 20. Identification of Two Spheres. Relative L2 error of the velocity for the full data test case of two spheres.

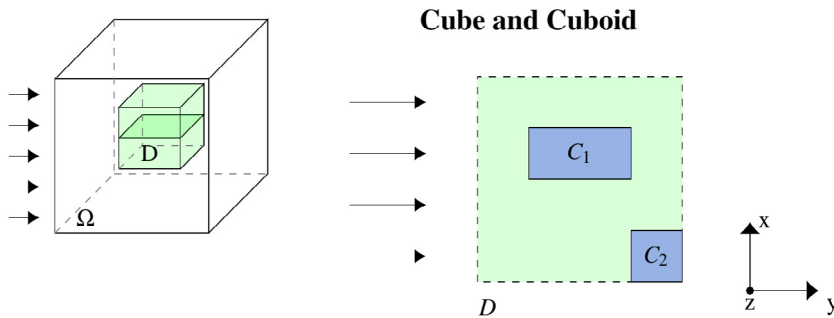


Fig. 21. Object Identification: Cube and Cuboid (Scheme). A small cube C_1 of side length 0.1 m (blue) is placed in an upper corner of the design domain D (green), extending from (0.6, 0.6, 0.6) to (0.7, 0.7, 0.7). An additional cuboid C_2 (blue) of side length (0.1 m, 0.2 m, 0.2 m) extending from (0.4, 0.4, 0.4) to (0.5, 0.6, 0.6) is placed in the design domain D (green). The geometry $\{C_1, C_2\} =: \Gamma$ is to be identified.

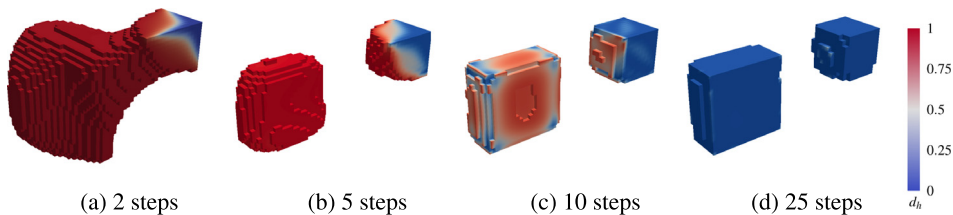


Fig. 22. Object Identification: Cube and Cuboid (Results). Optimisation results after 2, 5, 10 and 25 steps are visualised for the test case described in Fig. 21.

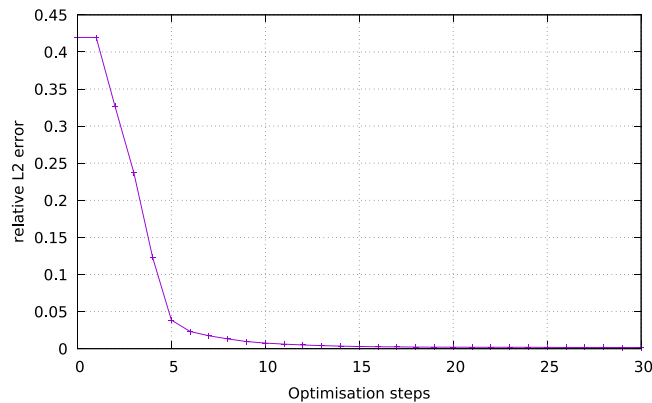


Fig. 23. Identification of a Cube and a Cuboid. Relative L2 Error of the velocity for the full data test case of a cube and a cuboid.

of decreasing flow data being available. In future, improvements, as for example additional conditions on the volume, are to be found that ensure the uniqueness of the identified objects. Further, a combination with MRI measurements can be considered. For it, MRI data could serve as the given velocity profiles. A significant reduction of the noise in the flow field is expected as well as an identification of the flow domain, i.e. a new segmentation approach is expected, which projects the measured flow field very close to the physical one.

Acknowledgements

The authors would like to thank Tim Dornieden and Lukas Baron from KIT (Karlsruhe, Germany) for their contributions to ALBM as well as Willy Dörfler and Hermann Nirschl, both from KIT, for providing the work facilities and excellent supervision. The authors would also like to thank the Steinbuch Centre for Computing at KIT for allowing access to their high performance computer FH1, where most computations have been performed.

References

- [1] B. Chopard, M. Droz, Cellular Automata Modeling of Physical Systems, Cambridge University Press, 1998. <http://dx.doi.org/10.1017/CBO9780511549755>.
- [2] T. Henn, G. Thäter, W. Dörfler, H. Nirschl, M. Krause, Parallel dilute particulate flow simulations in the human nasal cavity, *Comput. & Fluids* 124 (2016) 197–207. <http://dx.doi.org/10.1016/j.compfluid.2015.08.002>.
- [3] V. Heuveline, F. Strauss, Shape optimization towards stability in constrained hydrodynamic systems, *J. Comput. Phys.* 228 (2009) 938–951.
- [4] G. Pingan, A. Evgrafov, K. Maute, Topology optimization of flow domains using the lattice Boltzmann method, *Struct. Multidiscip. Optim.* 34 (6) (2007) 507–524. <http://dx.doi.org/10.1007/s00158-007-0105-7>.
- [5] G. Pingan, A. Evgrafov, K. Maute, A parallel Schur complement solver for the solution of the adjoint steady-state lattice Boltzmann equations: application to design optimisation, *Int. J. Comput. Fluid Dyn.* 22 (7) (2008) 457–464.
- [6] G. Pingan, A. Evgrafov, K. Maute, Adjoint parameter sensitivity analysis for the hydrodynamic lattice Boltzmann method with applications to design optimization, *Comput. & Fluids* 38 (4) (2009) 910–923. <http://dx.doi.org/10.1016/j.compfluid.2008.10.002>. URL: <http://www.sciencedirect.com/science/article/B6V26-4TTMJN3-1/2/16383afe088243863f7bc5f569da1279>.
- [7] S. Kreisli, G. Pingan, K. Maute, Topology optimization for unsteady flow, *Internat. J. Numer. Methods Engrg.* 87 (13) (2011) 1229–1253. <http://dx.doi.org/10.1002/nme.3151>.
- [8] M.M. Tekitek, M. Bouzidi, F. Dubois, P. Lallemand, Adjoint lattice Boltzmann equation for parameter identification, *Comput. & Fluids* 35 (2006) 805–813. <http://dx.doi.org/10.1016/j.compfluid.2005.07.015>.
- [9] M.J. Krause, G. Thäter, V. Heuveline, Adjoint-based fluid flow control and optimisation with lattice Boltzmann methods, in: *Mesoscopic Methods in Engineering and Science*, *Comput. Math. Appl.* 65 (6) (2013) 945–960. <http://dx.doi.org/10.1016/j.camwa.2012.08.007>.
- [10] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with PDE Constraints, vol. 23, Springer Science & Business Media, 2008. <http://dx.doi.org/10.1007/978-1-4020-8839-1>.
- [11] M.D. Gunzburger, Perspectives in Flow Control and Optimization, vol. 5, SIAM, 2003.
- [12] 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, *Comput. Math. Appl.* 68 (10) (2014) 1374–1392. <http://dx.doi.org/10.1016/j.camwa.2014.09.002>.
- [13] L. Laniewski-Wollk, J. Rokicki, Adjoint Lattice Boltzmann for topology optimization on multi-GPU architecture, *Comput. Math. Appl.* 71 (3) (2016) 833–848. <http://dx.doi.org/10.1016/j.camwa.2015.12.043>.
- [14] K. Yaji, T. Yamada, M. Yoshino, T. Matsumoto, K. Izui, S. Nishiwaki, Topology optimization using the lattice Boltzmann method incorporating level set boundary expressions, *J. Comput. Phys.* 274 (2014) 158–181. <http://dx.doi.org/10.1016/j.jcp.2014.06.004>.
- [15] K. Yonekura, Y. Kanno, A flow topology optimization method for steady state flow using transient information of flow field solved by lattice Boltzmann method, *Struct. Multidiscip. Optim.* 51 (1) (2015) 159–172. <http://dx.doi.org/10.1007/s00158-014-1123-x>.
- [16] 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. http://dx.doi.org/10.1007/978-3-319-47066-5_23.

- [17] R.H. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, *SIAM J. Sci. Comput.* 16 (5) (1995) 1190–1208. <http://dx.doi.org/10.1137/0916069>.
- [18] M.D. Gunzburger, Perspectives In Flow Control and Optimization, in: *Advances in Design and Control*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002. URL: <http://www.ulb.tu-darmstadt.de/toocs/129935174.pdf>.
- [19] 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.
- [20] L. Saint-Raymond, From the BGK model to the Navier-Stokes equations, *Ann. Sci. Éc. Norm. Supér.* 36 (2) (2003) 271–317. [http://dx.doi.org/10.1016/S0012-9593\(03\)00010-7](http://dx.doi.org/10.1016/S0012-9593(03)00010-7). URL: <http://www.sciencedirect.com/science/article/B6VKH-48HS9DK-5/2/4b7102c9ed9f501112dc9b08b7c9ae3d>.
- [21] M.A.A. Spaid, F.R. Phelan Jr., Lattice Boltzmann methods for modeling microscale flow in fibrous porous media, *Phys. Fluids* 9 (9) (1997) 2468–2474. <http://dx.doi.org/10.1063/1.869392>.
- [22] C. Geiger, C. Kanzow, Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben, in: *Springer-Lehrbuch*, Springer, Berlin, 1999. URL: <http://swbplus.bsz-bw.de/bsz080178243inh.htm>, <http://swbplus.bsz-bw.de/bsz080178243cov.htm>.
- [23] M.J. Krause, F. Klemens, T. Henn, R. Trunk, H. Nirschl, Particle flow simulations with homogenised lattice Boltzmann methods, *Particuology* 34 (Supplement C) (2017) 1–13. <http://dx.doi.org/10.1016/j.partic.2016.11.001>.
- [24] M.C. Sukop, D.T. Thorne, *Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers*, Springer, 2006. <http://dx.doi.org/10.1007/978-3-540-27982-2>.
- [25] D. Hänel, *Molekulare Gasdynamik*, Springer, 2004.
- [26] 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), Universität Karlsruhe (TH), Kaiserstraße 12, 76131 Karlsruhe, Germany, 2010. URL: <http://digbib.ubka.uni-karlsruhe.de/volltexte/1000019768>, <http://digbib.ubka.uni-karlsruhe.de/volltexte/1000019768>.
- [27] D.A. Wolf-Gladrow, Lattice-Gas, Cellular Automata and Lattice Boltzmann Models, An Introduction, in: *Lecture Notes in Mathematics*, Springer, Heidelberg, Berlin, 2000.
- [28] V. Heuveline, M. Krause, J. Latt, Towards a hybrid parallelization of lattice Boltzmann methods, *Comput. Math. Appl.* 58 (2009) 1071–1080. <http://dx.doi.org/10.1016/j.camwa.2009.04.001>.