Computers and Mathematics with Applications 68 (2014) 1374–1392

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Discrete adjoint sensitivity analysis for fluid flow topology optimization based on the generalized lattice Boltzmann method

Geng Liu [a,b], Martin Geier [c], Zhenyu Liu [a,*], Manfred Krafczyk [c], Tao Chen [a]

[a] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, 130033 Changchun, Jilin, PR China

[b] University of Chinese Academy of Sciences, 100049 Beijing, PR China

[c] Institute for Computational Modelling in Civil Engineering, Technische Universität Braunschweig, Pockelsstr. 3, 38106 Braunschweig, Germany

ARTICLE INFO

Article history:
Received 18 July 2013
Received in revised form 2 September 2014
Accepted 7 September 2014
Available online 29 September 2014

Keywords:
Discrete adjoint analysis
Lattice Boltzmann method
Topology optimization

ABSTRACT

A discrete adjoint sensitivity analysis for fluid flow topology optimization based on the lattice Boltzmann method (LBM) with multiple-relaxation-times (MRT) is developed. The lattice Boltzmann fluid solver is supplemented by a porosity model using a Darcy force. The continuous transition from fluid to solid facilitates a gradient based optimization process of the design topology of fluidic channels. The adjoint LBM equation, which is used to compute the gradient of the optimization objective with respect to the design variables, is derived in moment space and found to be as simple as the original LBM. The moment based spatial momentum derivatives used to express the discrete objective functional (cost function) have the advantage that the local stress tensor is a local quantity avoiding the numerical computation of gradients of the discrete velocity field. This is particularly useful if dissipation is a design criterion as demonstrated in this paper. The method is validated by a detailed comparison with results obtained by Borrvall et al. for Stokes flow. While their approach is only valid for Stokes flow (i.e. very low Reynolds numbers) our approach in its present form can in principle be applied for flows of different Reynolds numbers just like the Navier–Stokes equation based approaches. This point is demonstrated with a bending pipe example for various Reynolds numbers.

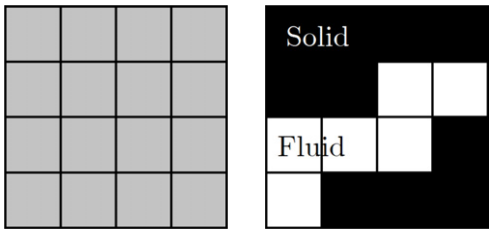
© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

A major goal of computational engineering is to automatically determine optimal designs for devices performing specified tasks. In addition to viable simulation tools for the physical phenomenon at hand, the formulation of the desired optimal condition in form of a cost function is a pre-requirement for achieving this goal. If the cost function is known and if it is sufficiently smooth with respect to the variation of design parameters it is in principle possible to approach a (locally) optimal design by following the gradient of the cost function with respect to the design variables. The physical model appears as a side condition to this process. In this paper we introduce an extended LBM scheme for the optimization of fluidic channels. The design variables correspond to the layout of the boundary conditions and the physical side condition is given by the Navier–Stokes equation. The choice of the design variables plays a key role in this procedure. The most general approach is to allow each grid point of the domain to be either solid or fluid. This requires one design variable for each grid

* Corresponding author.
E-mail address: liuzy@ciomp.ac.cn (Z. Liu).

http://dx.doi.org/10.1016/j.camwa.2014.09.002
0898-1221/© 2014 Elsevier Ltd. All rights reserved.



(a) Pre-optimized. (b) Post-optimized.

Fig. 1. Topology evolution of density type method.

point in the domain implying that the number of design variables is of the same order as the total number of degrees of freedom of the simulation itself. This general approach is called topology optimization because it allows the topology of the flow field to be changed by e.g. generating new junctions connecting different fluid subdomains. Topology optimization for fluidic devices has been introduced using two different concepts, namely the porosity or density type method (Fig. 1) where the flow field is modelled as a porous medium and the local permeability is a design variable [1–5], and the level-set method where the boundary is represented by the contour of a signed distance function [6–11]. In the current paper we consider the density type method implemented within an LBM framework for solving the fluid field and the adjoint equations.

The cost function $J = J(\varphi, \gamma)$ is a function of the physical variables φ (in our case the hydrodynamic variables such as velocity, pressure, and shear rate) and the design variables γ (in our case the local density of the porous solid material). The value of the cost function can be determined by solving the so-called forward problem which is the usual physical simulation of the device. For performing the optimization either with conjugate gradient methods or with the method of moving asymptotes (MMA) [12] the gradient of J with respect to γ has to be computed. Since the size of the field γ is comparable to the total number of degrees of freedom of the numerical setup for the momentum and mass conservation equations, it is essentially intractable to compute the gradient directly. Instead, we propose to use the adjoint sensitivity analysis. We write the numerical forward problem including initial and boundary conditions in the form:

$$e(\varphi, \gamma) = 0. \tag{1}$$

The Lagrangian of the optimization model is then:

$$\hat{J} = J(\varphi, \gamma) + \lambda e(\varphi, \gamma) \tag{2}$$

where λ represents the Lagrangian parameter. Note that λ, φ, and γ are all fields that potentially vary in space and time. We further note that $\hat{J} = J$ since $e(\varphi, \gamma) = 0$. Taking the variations of $\hat{J}$ and J respectively yields:

$$\delta\hat{J} = \delta\varphi \left(\frac{\partial J}{\partial \varphi} + \frac{\partial e(\varphi, \gamma)}{\partial \varphi}\lambda + \frac{\partial \lambda}{\partial \varphi} e(\varphi, \gamma) \right) + \delta\gamma \left(\frac{\partial J}{\partial \gamma} + \frac{\partial e(\varphi, \gamma)}{\partial \gamma}\lambda + \frac{\partial \lambda}{\partial \gamma} e(\varphi, \gamma) \right) \tag{3}$$

$$\delta J = \delta\varphi \frac{\partial J}{\partial \varphi} + \delta\gamma \frac{\partial J}{\partial \gamma}. \tag{4}$$

Since the Lagrangian parameter is no explicit function of φ and γ we have:

$$\frac{\partial \lambda}{\partial \varphi} = 0; \qquad \frac{\partial \lambda}{\partial \gamma} = 0. \tag{5}$$

A (local) optimum can only exist where the gradient of the cost function with respect to the design variables vanishes:

$$\frac{\partial J}{\partial \gamma} = 0. \tag{6}$$

Further we have $\delta J = \delta\hat{J}$ and since the variations are arbitrary we can separate them to deduce the Karush–Kuhn–Tucker conditions [13]:

$$\text{primal equation:} \quad e(\varphi, \gamma) = 0 \tag{7}$$

$$\text{adjoint equation:} \quad \frac{\partial J}{\partial \varphi} + \frac{\partial e(\varphi, \gamma)}{\partial \varphi}\lambda = 0 \tag{8}$$

$$\text{sensitivity equation:} \quad \frac{\partial J}{\partial \gamma} + \frac{\partial e(\varphi, \gamma)}{\partial \gamma}\lambda = 0 \tag{9}$$

where λ becomes the adjoint variable and is typically a function of space and time. For any given design variable γ the gradient of the cost function (sensitivity) can be obtained by first solving the forward problem to obtain φ. With φ the adjoint variable λ is obtained by solving the adjoint equation. Finally the sensitivity equation is solved for the gradient using φ and λ.

The adjoint sensitivity analysis in the context of fluidic topology optimization has been used by Deng et al. [5,14], Zhou et al. [9] and Kreissl et al. [11,15] using a finite element method for the Navier–Stokes problem. The adjoint analysis in the context of the lattice Boltzmann method was pioneered by Tekitek et al. for the optimization of relaxation rates of a multiple-relaxation-time LBM [16]. Pingen [17–22] conducted a discrete adjoint sensitivity analysis based on LBM for various stationary topology optimization problems. In Pingen's approach the adjoint algorithm is obtained by using a Jacobian matrix. The constraint of the corresponding optimization problem was implemented by equating two subsequent time steps and hence imposing a steady state. Based on Pingen's approach, Kirk et al. did some extension in the field of transient problems [23]. In our study we apply the variational method directly on the lattice Boltzmann algorithm. Our goal functional can also be an integral over time without assuming a steady state condition. Thus, while targeting on similar problems, Pingen's approach differs both conceptually as well as algorithmically from the approach we derive below. Pingen did not actually derive an adjoint lattice Boltzmann method (ALBM). Instead of updating the adjoint variables of each discrete spatial positions, his approach concentrates on updating the whole adjoint variable vector by building up a matrix as a coefficient in a linear adjoint equation set (or a recursive expression in the transient case), which solves adjoint variables of specific positions and lattice velocities, and needs additional linear or/and matrix solvers. Krause presented a continuous adjoint sensitivity analysis for flow control and optimization based on the lattice Boltzmann equation using a Bhatnagar–Gross–Krook (BGK) collision operator [24]. Starting from the continuous forward equation his derivation is rather complex and to the best of our knowledge has not been applied to topology optimization. Krause's approach is also limited to BGK because a more generalized LBM cannot be traced back to a continuous form. The works of Pingen and Krause are based on the adjoint of the distribution function. This also requires to express the cost function in terms of the distribution function. Although in [22] Pingen provided a MRT approach, he was focusing on the accuracy and stability of the forward problem of simulating flow with multiple components. His main idea of deriving the adjoint algorithm does not use moments and he still has to transform the macroscopic quantities in his cost function into distribution space when computing the adjoint problem.

Moments and distributions are mutually exchangeable mathematical representations of the same state. Therefore, deriving the adjoint in moment space or in distribution space should be equivalent. However, the moment approach leads to a much simpler derivation and potentially to a more efficient implementation of the adjoint LBM since most cost functions are more efficiently and naturally stated as functions of macroscopic variables like velocity, pressure, and shear rate rather than in the form of discrete distributions. All these variables are moments of the distributions. It is beneficial for the purpose of optimization that also some spatial derivatives of velocity are presented as moments of the distributions as can be shown with various methods of asymptotic analysis [25–27]. The cost function can depend on any macroscopic quantity or any spatial or temporal derivative of a macroscopic quantity. In general it is not required that the same derivatives appear in the computation of the forward problem and additional finite differences can be required to obtained the required variables. In many practical cases the cost function depends on derivatives that are also required during the computation of the forward problem. In this case the LBM allows us to recycle the information from the forward problem in the cost function. In the moment approach the derivatives computed for the forward problem can be easily selected from the appropriate moments. Without this knowledge we would have to use additional finite differences for calculating, for example, dissipation. A distribution based approach could use the same information to avoid the additional finite difference but it had to compute the moments in the process. Since moments of the discrete distribution functions can be easily linked to fluid dynamic variables and their spatial derivatives, a simplification of the sensitivity analysis is obtained when the adjoint of the collision operator is derived in moment space. In addition to this simplification two more advantages arise: first, it has been observed that the moment based multiple-relaxation-time lattice Boltzmann method is in general much more stable and accurate than the single-relaxation-time collision operator at least for the forward problem [28] and it can be used for studying physically more complex fluids such as mixtures with non unity Schmidt number as done by Pingen et al. [22] in the context of fluid topology optimization. Second, as mentioned above the computation of dissipation can be based on local moments alone and does not require additional finite differences. As minimization of dissipation is a common design goal in engineering this property is an advantage of our approach. In this paper we derive the adjoint lattice Boltzmann equation in moment space. Only the streaming step requires the consideration of the adjoint distribution function.

The interesting aspect of solving the fluid flow topology optimization problem with ALBM is that solving the adjoint problem is algorithmically almost identical with solving the forward problem. The adjoint lattice Boltzmann equation is solved on parallel hardware using exactly the same algorithms that have been developed for the usual LBM. In this respect our ALBM approach is similar to the discrete realization of an adjoint for an explicit time marching finite difference scheme. It is not the purpose of this paper to assess the exact performance of our particular implementation but it is evident that our approach automatically benefits from all ongoing computational improvements of the LBM since the algorithm to solve the adjoint LBM is identical to the one solving the forward LBM. Our approach can be used for steady and unsteady problems. Although we have experience in solving unsteady problems in topology optimization with a Navier–Stokes based finite element method [5], the ability to solve these problems with LBM might be limited by memory requirements since the complete time history of the forward problem has to be recorded [23]. In the case of steady problems no such limitation applies as will be explained in Section 4. Solving a steady problem with a time marching method is potentially less efficient than using an implicit method but benchmarks have shown that the difference in computational time is still competitive [29]. In the case of an iterative optimization process, as used here, reaching a steady state is only costly in the first iteration since all following iterations can be started from the steady solution of the previous iteration. The requirement to store the

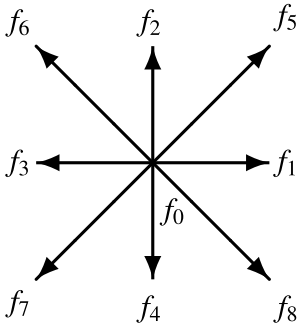


Fig. 2. D2Q9 lattice.

history of the forward problem is not a specific disadvantage of the adjoint LBM since it applies also to any other unsteady sensitivity analyses such as approaches based on the Navier–Stokes equation [5,15]. Even though there might be obstacles other than memory consumption for solving unsteady problems with LBM, we still regard it as a foreseeable difficulty. Hence we discuss briefly several techniques to reduce the huge consumption of memory here. First of all, memory-reduced methods such as one shot (also called Simultaneous Analysis and Design (SAND)) [30] and checkpointing [31,32] are widely used in the implementation of time-dependent optimization problems. The SAND approach might work well in engineering applications, especially when the forward problem is slowly varying in time. The SAND method does save memory if applied to an unsteady state problem. Secondly, data compression as a programming trick is more and more popular as the relevant hardware and concepts develops. Third, Vergnault et al. has developed a time-reversal lattice Boltzmann method [33], which is deemed to be capable to solve the storage problem by re-simulating backward while the adjoint algorithm is processed. However, this approach is unstable for long times as the reversal of fluid dynamics is a fundamentally ill posed problem due to the negative viscosity involved. The method developed in this paper can be combined with all of the above approaches to facilitate time dependent optimization.

2. LBM based primal process solver

We demonstrate our method for a standard two dimensional multiple-relaxation-time LBM. The lattice Boltzmann approach can be interpreted as an explicit numerical solver for weakly compressible Navier–Stokes problems employing a discrete probability distribution function (PDF) f_i as primal variables. The PDFs are located on the grid nodes of a Cartesian grid, undergo subsequent modification by a local collision operator and move according to their discrete velocity $\boldsymbol{c}_i$ from node to node. Since the discrete velocity set (multiplied by the time step) is chosen to be congruent to the numerical grid, advection is exact and no interpolation of the PDFs is required. In this paper we consider all variables to be normalized such that the spacing between two lattices nodes $\delta_x = 1$ and the time step $\delta_t = 1$ are both unity. On each node the local distribution vector is transformed into moment space and each non-conserved moment is individually relaxed towards an equilibrium value before it is transformed back into distribution space. This collision step is followed by a new propagation or streaming step. The LBM update algorithm without body force can hence be written as:

$$\text{Collision:}\quad f_i^*(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) - \left[\boldsymbol{M}^{-1}\boldsymbol{S}\left(\boldsymbol{M}\boldsymbol{f}(\boldsymbol{x}, t) - \boldsymbol{m}^{\mathrm{eq}}(\boldsymbol{x}, t)\right)\right]_i \tag{10}$$

$$\text{Streaming:}\quad f_i(\boldsymbol{x} + \boldsymbol{c}_i, t + 1) = f_i^*(\boldsymbol{x}, t). \tag{11}$$

Here $\boldsymbol{c}_i$ is the discrete velocity of the distribution f_i. For the two dimensional nine velocity model used here (D2Q9 Fig. 2 [34]) the velocities are given by:

$$\{\boldsymbol{c}_i\} = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}, \quad i = 0, 1, \ldots, 8. \tag{12}$$

$\boldsymbol{M}$ is the transformation matrix chosen here as [35]:

$$\boldsymbol{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix}. \tag{13}$$

The relaxation rate matrix $\boldsymbol{S}$, the moment vector $\boldsymbol{m} = \boldsymbol{M}\boldsymbol{f}$ and the vector of equilibrium moments $\boldsymbol{m}^{\mathrm{eq}}$ are defined as:

$$\begin{aligned}
&\boldsymbol{S} = \mathrm{diag}(0, s_e, s_\varepsilon, 0, s_q, 0, s_q, s_\nu, s_\nu)\\
&\boldsymbol{m} = [\rho, e, \epsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy}]^{\mathrm{T}}\\
&\boldsymbol{m}^{\mathrm{eq}} = \rho[1, -2 + 3\boldsymbol{u}^2, 1 - 3\boldsymbol{u}^2, u_x, -u_x, u_y, -u_y, u_x^2 - u_y^2, u_x u_y]^{\mathrm{T}}.
\end{aligned} \tag{14}$$

The equilibrium moments depend on the three conserved moments mass ρ and momentum j_x and j_y. The velocity is given by $\boldsymbol{u} = [u_x, u_y]^{\mathrm{T}} = [j_x, j_y]^{\mathrm{T}}\rho^{-1}$. The conserved moments can be computed from the pre-collision distribution function:

$$\rho = \{\boldsymbol{M}\boldsymbol{f}\}_1, \qquad j_x = \{\boldsymbol{M}\boldsymbol{f}\}_4, \qquad j_y = \{\boldsymbol{M}\boldsymbol{f}\}_6. \tag{15}$$

The post-collision state of distributions $\boldsymbol{f}^*$ and the moment vectors $\boldsymbol{m}^*$ are indicated by the asterisk throughout this paper. The kinematic viscosity ν determines the relaxation rate s_ν:

$$\frac{1}{s_\nu} = 3\nu + \frac{1}{2}. \tag{16}$$

All other relaxation rates are free parameters and can be chosen in the range 0–2.

The optimization process requires a smooth transition between fluid and solid. We model this transition with a global immersed boundary method (porous medium method) where the solid imposes a Darcy force counteracting the flow. Momentum is no longer conserved locally when a force is applied. In the case of the porosity model the force is a function of the velocity and the velocity is naturally a function of the force. It is therefore convenient to apply a forcing model that uses the pre-forcing velocity for the collision so that no implicit dependence of the velocity on the force has to be considered. The Ladd–Verberg-I force model [36] has this property. It is defined by:

$$\bar{\boldsymbol{F}}^{(1)} = [0, 0, 0, F_x, -F_x, F_y, -F_y, 0, 0]^{\mathrm{T}} \tag{17}$$

$$\bar{\boldsymbol{F}}^{(2)} = [0, 6(F_x u + F_y v), -6(F_x u + F_y v), 0, 0, 0, 0, 2(F_x u - F_y v), F_y u + F_x v]^{\mathrm{T}} \tag{18}$$

where $\boldsymbol{F} = [F_x, F_y]^{\mathrm{T}}$ is the body force. The collision step with body force in moment space is written as:

$$\boldsymbol{m}^* = \boldsymbol{m} - \boldsymbol{S}\left(\boldsymbol{m} - \boldsymbol{m}^{\mathrm{eq}}\right) + \bar{\boldsymbol{F}}^{(1)} + (\boldsymbol{I} - \boldsymbol{S}/2)\,\bar{\boldsymbol{F}}^{(2)}. \tag{19}$$

This model uses the pre-collision velocity for relaxation and adds a second-order correction to the stress.

To facilitate the derivation of the adjoint lattice Boltzmann equation we restrict ourselves to so-called nodal boundary conditions that replace the collision operator by a boundary collision operator. Node based boundary conditions exist for all standard boundaries such as no-slip, velocity, and pressure boundary conditions. In this study the no-slip boundary Γ_b is implemented using the node-based simple bounce back method:

$$\boldsymbol{m}^*|_{\boldsymbol{x}\in\Gamma_b} = \boldsymbol{B}\boldsymbol{m} \tag{20}$$

where the bounce back operator in moment space is given by

$$\boldsymbol{B} = \mathrm{diag}(1, 1, 1, -1, -1, -1, -1, 1, 1). \tag{21}$$

At the inlet Γ_{in}, fixed momentums are given:

$$\boldsymbol{j}|_{\boldsymbol{x}\in\Gamma_{\mathrm{in}}} = \boldsymbol{j}_{\mathrm{in}} \quad (\boldsymbol{j} = [j_x, j_y]). \tag{22}$$

At the outlet Γ_{out}, zero shear stress is given as:

$$(-p\boldsymbol{I} + \rho\nu\boldsymbol{\Pi})\,\boldsymbol{l}|_{\boldsymbol{x}\in\Gamma_{\mathrm{out}}} = \boldsymbol{0} \tag{23}$$

where $p = (\rho - 1)/3$ is the hydrodynamic pressure at the outlet, $\boldsymbol{\Pi} = \nabla\boldsymbol{u} + \nabla\boldsymbol{u}^{\mathrm{T}}$ is the strain rate tensor and $\boldsymbol{l}$ is the normal boundary vector.

3. Discrete adjoint analysis

We consider the cost function J to be the sum over local cost functions $G(\boldsymbol{m}, \gamma, \boldsymbol{x})$:

$$J = \sum_{t=t_0}^{t_1} \sum_{\boldsymbol{x}\in\mathrm{D}} G(\boldsymbol{m}, \gamma, \boldsymbol{x}) \tag{24}$$

where t_0 is the starting time, t_1 is the ending time and D is the domain (Fig. 3), respectively. The local cost function $G(\boldsymbol{m}, \gamma, \boldsymbol{x})$ defines the objective for the design task at hand.

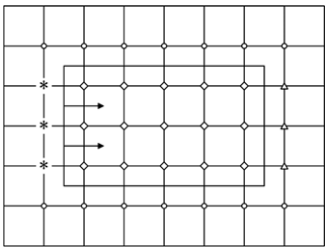


Fig. 3. Domain: ○—bounce back nodes, ∗—inlet nodes, ◇—fluid nodes, △—outlet nodes; the nodes without symbols belong to outer area.

Without loss of generality we assume that the simulation domain is periodic in all directions. Any finite domain can be mapped onto a sufficiently large periodic domain with internal boundary nodes. The lattice Boltzmann algorithm termed $e(\varphi, \gamma)$ is split into four steps:

$$\text{moment transform:} \quad \boldsymbol{e}_1 = \boldsymbol{M}\boldsymbol{f} - \boldsymbol{m} = \boldsymbol{0} \tag{25}$$

$$\text{collision:} \quad \boldsymbol{e}_2 = \boldsymbol{C}(\boldsymbol{m}, x, t, \gamma) - \boldsymbol{m}^* = \boldsymbol{0} \tag{26}$$

$$\text{inverse moment transform:} \quad \boldsymbol{e}_3 = \boldsymbol{M}^{-1}\boldsymbol{m}^* - \boldsymbol{f}^* = \boldsymbol{0} \tag{27}$$

$$\text{streaming:} \quad e_{4i} = f_i^*(\boldsymbol{x} - \boldsymbol{c}_i, t-1) - f_i(\boldsymbol{x}, t) = 0. \tag{28}$$

Here, $\boldsymbol{C}$ is a generalized collision operator representing all possible local update rules including those for boundary conditions. Some boundary conditions require interpolation and their collision operator thus depends on non-local moments $\boldsymbol{m}$. The transformation to and from moments does not need to be considered as an individual step in the algorithm but we find it more convenient to add it here. It leads to just another term in the sensitivity analysis when applying the chain rule but allows us to write the cost function directly in terms of macroscopic variables.

We introduce Lagrangian multipliers for each step:

$$e(\varphi, \gamma)\lambda = e(\boldsymbol{m}, \boldsymbol{m}^*, \boldsymbol{f}, \boldsymbol{f}^*, \gamma)\lambda = \sum_{t=t_0}^{t_1} \sum_{\boldsymbol{x}\in D} \boldsymbol{e}_1 \cdot \boldsymbol{n}^{*\mathrm{T}} + \boldsymbol{e}_2 \cdot \boldsymbol{n}^{\mathrm{T}} + \boldsymbol{e}_3 \cdot \boldsymbol{g}^{\mathrm{T}} + \boldsymbol{e}_4 \cdot \boldsymbol{g}^{*\mathrm{T}} \tag{29}$$

$\boldsymbol{n}, \boldsymbol{n}^*, \boldsymbol{g}, \boldsymbol{g}^*$ will be identified below as the adjoint pre-collision and post-collision moments and distributions respectively. Thus the Lagrangian functional becomes:

$$\hat{J} = \sum_{t=t_0}^{t_1} \sum_{\boldsymbol{x}\in D} G + \boldsymbol{e}_1 \cdot \boldsymbol{n}^{*\mathrm{T}} + \boldsymbol{e}_2 \cdot \boldsymbol{n}^{\mathrm{T}} + \boldsymbol{e}_3 \cdot \boldsymbol{g}^{\mathrm{T}} + \boldsymbol{e}_4 \cdot \boldsymbol{g}^{*\mathrm{T}}. \tag{30}$$

Following the procedure outlined in the introduction we take the variation of $\hat{J}$ which is known to be zero for the optimal solution:

$$\delta\hat{J} = 0 = \delta\gamma \frac{\partial \hat{J}}{\partial \gamma} + \delta\boldsymbol{m} \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{m}} + \delta\boldsymbol{m}^* \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{m}^*} + \delta\boldsymbol{f} \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{f}} + \delta\boldsymbol{f}^* \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{f}^*}. \tag{31}$$

Since the variations are independent from each other, the sum can only be zero if and only if all individual summands are zero:

$$\delta\gamma \frac{\partial \hat{J}}{\partial \gamma} = 0, \qquad \delta\boldsymbol{m} \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{m}} = 0, \qquad \delta\boldsymbol{m}^* \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{m}^*} = 0, \qquad \delta\boldsymbol{f} \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{f}} = 0, \qquad \delta\boldsymbol{f}^* \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{f}^*} = 0. \tag{32}$$

The first equation is the actual optimality condition of the objective. The other equations are used to derive the adjoint lattice Boltzmann algorithm. The adjoint collision is obtained from:

$$\delta\boldsymbol{m} \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{m}} = \sum_{t=t_0}^{t_1} \sum_{\boldsymbol{x}\in D} \delta\boldsymbol{m} \cdot \left(\frac{\partial G}{\partial \boldsymbol{m}} - \boldsymbol{n}^{*\mathrm{T}} + \boldsymbol{n}^{\mathrm{T}} \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{m}} \right) = 0 \tag{33}$$

which implies:

$$\boldsymbol{n}^* = \left(\frac{\partial G}{\partial \boldsymbol{m}} \right)^{\mathrm{T}} + \left(\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{m}} \right)^{\mathrm{T}} \cdot \boldsymbol{n}. \tag{34}$$

Next, the adjoint moment transform is obtained:

$$\delta \boldsymbol{m}^* \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{m}^*} = \sum_{t=t_0}^{t_1} \sum_{\boldsymbol{x} \in \mathrm{D}} \delta \boldsymbol{m}^* \cdot \left(-\boldsymbol{n}^{\mathrm{T}} + \boldsymbol{g}^{\mathrm{T}} \boldsymbol{M}^{-1}\right) = 0 \tag{35}$$

$$\boldsymbol{n} = \boldsymbol{M}^{-\mathrm{T}} \boldsymbol{g}. \tag{36}$$

It is interesting to note that the transformation matrix in the adjoint LBM is the transposed inverse of the transformation matrix of the original LBM (similar discoveries were made by Tekitek et al. in [16]). We also obtain the transformation for the post-collision state:

$$\delta \boldsymbol{f} \cdot \frac{\partial \hat{J}}{\partial \boldsymbol{f}} = \sum_{t=t_0}^{t_1} \sum_{\boldsymbol{x} \in \mathrm{D}} \delta \boldsymbol{f} \cdot \left(\boldsymbol{n}^{*\mathrm{T}} \boldsymbol{M} - \boldsymbol{g}^{*\mathrm{T}}\right) = 0 \tag{37}$$

$$\boldsymbol{g}^* = \boldsymbol{M}^{\mathrm{T}} \boldsymbol{n}^*. \tag{38}$$

Finally, the adjoint streaming is obtained from the following equation:

$$\delta f_i^* \cdot \frac{\partial \hat{J}}{\partial f_i^*} = \sum_{t=t_0}^{t_1} \sum_{\boldsymbol{x} \in \mathrm{D}} \left[-\delta f_i^*(\boldsymbol{x}, t) g^{\mathrm{T}}(x, t) + \delta f_i^*(\boldsymbol{x} - \boldsymbol{c}_i, t-1) g_i^{*\mathrm{T}}(x, t)\right] = 0 \tag{39}$$

$$\begin{aligned} &= -\sum_{t=t_0}^{t_1} \sum_{\boldsymbol{x} \in \mathrm{D}} \delta f_i^*(\boldsymbol{x}, t) \left[g^{\mathrm{T}}(x, t) - g_i^{*\mathrm{T}}(\boldsymbol{x} + \boldsymbol{c}_i, t+1)\right] - \sum_{\boldsymbol{x} \in \mathrm{D}} \delta f_i^*(\boldsymbol{x}, t_1) g_i^{*\mathrm{T}}(\boldsymbol{x} + \boldsymbol{c}_i, t_1 + 1) \\ &\quad + \sum_{\boldsymbol{x} \in \mathrm{D}} \delta f_i^*(\boldsymbol{x}, t_0 - 1) g_i^{*\mathrm{T}}(\boldsymbol{x} + \boldsymbol{c}_i, t_0). \end{aligned} \tag{40}$$

Since we assumed without loss of generality that the simulation domain is periodic we do not need to split the sum in D when we shift the adjoint distributions and the variations of the primal distributions in space. Also, since we put all boundary conditions into the generalized collision operator, streaming is strictly the same for all nodes. Shifting the time from the variation of the distribution to the adjoint distribution adds a terminal value of $\boldsymbol{g}^*$ to the problem. However, since the variation $\delta \boldsymbol{f}^*$ is arbitrary this terminal value has to be zero so that the sum becomes zero and the terminal condition for the adjoint problem reads:

$$g_i^*(\boldsymbol{x} + c_i, t_1 + 1) = 0. \tag{41}$$

The adjoint streaming step can then be written in a more conventional form:

$$g_i(\boldsymbol{x} - \boldsymbol{c}, t - 1) = g_i^*(\boldsymbol{x}, t). \tag{42}$$

The fact that the time variable runs in negative direction (as expected) in the adjoint problem explains this equation very well. It is also seen that the adjoint distributions propagate in opposite direction from the original LBM (Eq. (11)). This phenomenon was first observed by Tekitek et al. in [16], and proved again in Krause's continuous approach [24]. It is also interesting to note that equations for the post-collision adjoint moments and distributions $\boldsymbol{n}^*$ and $\boldsymbol{g}^*$ arise from the variations of the pre-collision primal moments and distributions $\delta \boldsymbol{m}$ and $\delta \boldsymbol{f}$ and vice versa. So both the direction of the time and the order of streaming and collision is reversed in the adjoint problem.

The adjoint collision consists of contributions which directly depend on the cost function and a general part that is independent from the cost function. The independent part is obtained when we take the actual multiple-relaxation-time collision operator into consideration:

$$\boldsymbol{C} = \boldsymbol{m} - \boldsymbol{S}\left(\boldsymbol{m} - \boldsymbol{m}^{\mathrm{eq}}\right) + \bar{\boldsymbol{F}}^{(1)} + \left(\boldsymbol{I} - \frac{\boldsymbol{S}}{2}\right) \bar{\boldsymbol{F}}^{(2)} \tag{43}$$

$$\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{m}} = \boldsymbol{S}_A = \boldsymbol{I} - \boldsymbol{S}\left(\boldsymbol{I} - \frac{\partial \boldsymbol{m}^{\mathrm{eq}}}{\partial \boldsymbol{m}}\right) + \frac{\partial \bar{\boldsymbol{F}}^{(1)}}{\partial \boldsymbol{m}} + \left(\boldsymbol{I} - \frac{\boldsymbol{S}}{2}\right) \frac{\partial \bar{\boldsymbol{F}}^{(2)}}{\partial \boldsymbol{m}}. \tag{44}$$

We call $\boldsymbol{S}_A$ the adjoint collision matrix. For steady problems, it is a constant matrix which is available after the primal fluid field has been solved.

Since we did not lose generality in Eq. (34), we know that this form is suitable to all nodes that represent various physical attributes in the domain. In this paper we do not consider objectives on the boundary nodes, therefore the first part of Eq. (34) that is related to the objective does not exist on boundary nodes. However, if a specific objective is related to the boundary information, adding this part is straightforward. The second part of Eq. (34) is again related to the primal collision. Collision here means algorithms other than streaming. Particularly for the boundary nodes, collision can be node based bounce back

or other post streaming boundary treatments. The adjoint of the node-based simple bounce back boundary condition is just node-based simple bounce back:

$$\frac{\partial \boldsymbol{C}_b}{\partial \boldsymbol{m}} = \boldsymbol{B}. \tag{45}$$

In the LBM, momentum boundary conditions are treated as bounce back plus a constant amount of momentum. The constant momentum has zero adjoint. The adjoint of a momentum boundary condition is hence plain bounce back. The treatment of pressure boundary conditions is more complicated, and a thorough discussion of their adjoint in the LBM would justify a paper on its own. In the current study we use the *a posteriori* momentum at the outlet to map the pressure boundary condition onto a momentum boundary condition. The momentum at the outlet is known for all time steps after running the forward problem. Replacing the pressure boundary condition at the outlet by a momentum boundary condition using the *a posteriori* momentums leads to the same numerical solution. This *a posteriori* momentum boundary condition has bounce back as adjoint. Since we can, in principle, replace the pressure boundary condition by this *a posteriori* momentum boundary condition it is also possible to use bounce back as the adjoint of the pressure boundary condition. This will also be evident, at least for steady problems, from the numerical results below where all adjoint boundaries were treated with bounce back.

4. Sensitivity evaluation

In the following, we will only consider examples where the cost function vanishes at the boundaries. Hence, we compute the discrete sensitivity only for bulk nodes:

$$\frac{\mathrm{D}J}{\mathrm{D}\gamma(\boldsymbol{x})} = \sum_{t=t_0}^{t_1} \left[\frac{\partial G}{\partial \gamma(\boldsymbol{x})} + \left(\frac{\partial \boldsymbol{C}}{\partial \gamma(\boldsymbol{x})} \right)^{\mathrm{T}} \cdot \boldsymbol{n}(\boldsymbol{x}) \right] \tag{46}$$

where

$$\frac{\partial \boldsymbol{C}}{\partial \gamma(\boldsymbol{x})} = \frac{\partial \bar{\boldsymbol{F}}^{(1)}(\boldsymbol{x})}{\partial \gamma(\boldsymbol{x})} + \left(\boldsymbol{I} - \frac{\boldsymbol{S}}{2} \right) \frac{\partial \bar{\boldsymbol{F}}^{(2)}(\boldsymbol{x})}{\partial \gamma(\boldsymbol{x})} \tag{47}$$

and $\boldsymbol{n}$ is obtained via advancing the above adjoint process.

The most demanding task in calculating the adjoint variable $\boldsymbol{n}$ is the recording of the complete flow field for all time steps of the forward problem. Thus, for many practical problems of engineering interest the corresponding memory requirements will be prohibitive if all time steps are recorded directly. Griewank [31] proposed a checkpointing technique called "revolving" that reduces the required amount of memory by repeatedly computing the time steps between stored checkpoints. Both memory requirement and the number of necessary repeats of the forward problem grow at least logarithmically with the required number of time steps.

Fortunately, the situation is different for steady state problems. In a steady forward problem the flow field does not change in time after it is converged. We obtain the steady state similar to Krause [24] by checking for the result not to change anymore. The steady forward solution is then used for all time steps of the adjoint simulation. The adjoint simulation is run to steady state by checking for no change in the adjoint variables and the gradient is computed from the final step of the adjoint simulation. The adjoint problem always reaches a steady state since it is linear and its side conditions and boundary conditions are constant. The cost of the adjoint problem is hence comparable to the forward problem in terms of computational and memory requirements.

5. Cost function

The cost function G_{fluid} might have different functional forms depending on the location in space and time. In this example we assume that the cost function depends only on whether it is evaluated on a fluid node or on a boundary node. For fluid, inlet, outlet, and bounce-back nodes we distinguish the cost functions G_{fluid}, G_{in}, G_{out}, and G_b. In the current study we assume $G_{\text{in}} = G_{\text{out}} = G_b = 0$. At the regular fluid nodes we impose minimum dissipation as an objective. In two dimensions the dissipation can be expressed as:

$$J = \sum_{t=t_0}^{t_1} \sum_{\boldsymbol{x} \in \Omega} G \tag{48}$$

$$G_{\text{fluid}} = \nu\rho \left[2\left(\partial_x u_x\right)^2 + 2\left(\partial_y u_y\right)^2 + \left(\partial_x u_y + \partial_y u_x\right)^2 \right] - \boldsymbol{F} \cdot \boldsymbol{u}. \tag{49}$$

Minimum energy dissipation as a goal usually provides the designers with unobstructed channels. At the same time dissipation is also an interesting objective from the theoretical point of view as it typically requires the computation of spatial

derivatives of the velocity. In the lattice Boltzmann method the required derivatives are functions of the moments [37,38] as can be shown by Chapman–Enskog expansion [25], asymptotic analysis [26], or Taylor expansion [27].

$$\partial_x u_x + \partial_y u_y = \frac{s_e}{2\rho}\left(e^{\mathrm{eq}} - e\right) \tag{50}$$

$$\partial_x u_x - \partial_y u_y = \frac{3s_\nu}{2\rho}\left(p_{xx}^{\mathrm{eq}} - p_{xx}\right) \tag{51}$$

$$\partial_x u_y + \partial_y u_x = \frac{3s_\nu}{\rho}\left(p_{xy}^{\mathrm{eq}} - p_{xy}\right). \tag{52}$$

Thus dissipation can be written as a local function of the moments. No additional finite differences are required in the moment representation.

The body force is introduced to counteract the flow:

$$\boldsymbol{F} = -\alpha(\gamma)\rho\boldsymbol{u}. \tag{53}$$

This force is actually a resistance for the local fluid flow with a resistance coefficient related to the design variable. The coefficient α is expressed under the rule of Rational Approximation of Material Properties (RAMP [39]) as:

$$\alpha = \alpha_{\max}\frac{q(1-\gamma)}{q+\gamma} \tag{54}$$

where $\alpha_{\max}$ and q contribute to the solid criteria [3,4], the numbers are selected as a trade-off between the permeation caused by the post-optimized immersed boundary and the avoidance of obtaining a local optimum. In this paper q is fixed to 1.

We are using continuous design variables to describe solid (0) and fluid (1). This approach allows for the existence of grey areas, values between zero and one, that have no physical equivalence. In the converged design topology, grey areas should be removed as completely as possible. Experiments show that when dissipation is the objective and the selection of $\alpha_{\max}$ is rational, one would obtain a topology with very few intermediate material nodes if no mapping methods are applied to the raw design variables. The downside of not using mapping is that the optimization rapidly converges to clear solid/fluid design which is often a local optimum with inferior performance [3]. Through RAMP the design variables are mapped to coefficients with non-linear weights to counteract the strong non-linearity in the optimization variables on the flow solution and the dissipation. Basically this reduces the resistance of the grey areas. In this way the topology can converge within fewer optimizing steps. Smaller values of q make the process more efficient while at the same time smaller values of q allow for larger grey areas. The other important parameter in RAMP is $\alpha_{\max}$. Physically this parameter should be infinite because solids do not actually allow any fluid transport. Since infinity is not an admissible value in a numerical approach, $\alpha_{\max}$ should be as large as possible. Still, a too large $\alpha_{\max}$ with an inappropriate q might cause difficulties in solving the primal and adjoint problems and hide the reasonable topologies. Thus, the selection of this parameter is sometimes a compromise. The choice often depends on the domain size, the Reynolds number, and especially the lattice viscosity in the LBM. We demonstrate the influence of $\alpha_{\max}$ in the examples below.

Inserting the derivatives (Eqs. (50)–(52)) and the body force (Eq. (53)) into the local cost function (Eq. (49)) and taking the derivatives with respect to the moments we obtain:

$$\begin{aligned}\frac{\partial G_{\mathrm{fluid}}}{\partial \rho} &= 4\nu s_e^2 - 27\nu\left(4s_e^2 + s_\nu^2\right)\left(j_x^2 + j_y^2\right)^2/\left(4\rho^4\right) + 3\nu\left\{4s_e^2 e\left(j_x^2 + j_y^2\right) + 3s_\nu^2\left[p_{xx}\left(j_x^2 - j_y^2\right) + 4j_x j_y p_{xy}\right]\right\}/\rho^3 \\ &\quad - \nu\left\{4s_e^2\left[e^2 - 12\left(j_x^2 + j_y^2\right)\right] + 9s_\nu^2\left(p_{xx}^2 + 4p_{xy}^2\right)\right\}/\left(4\rho^2\right) + \left(F_x j_x + F_y j_y\right)/\rho^2\end{aligned} \tag{55}$$

$$\frac{\partial G_{\mathrm{fluid}}}{\partial e} = 2\nu s_e^2\left[-3\left(j_x^2 + j_y^2\right) + \rho\left(e + 2\rho\right)\right]/\rho^2 \tag{56}$$

$$\frac{\partial G_{\mathrm{fluid}}}{\partial j_x} = 9\nu j_x\left(4s_e^2 + s_\nu^2\right)\left(j_x^2 + j_y^2\right)/\rho^3 - 3\nu\left[4s_e^2 e j_x + 3s_\nu^2\left(j_x p_{xx} + 2j_y p_{xy}\right)\right]/\rho^2 - \left(F_x + 24\nu s_e^2 j_x + j_x\frac{\partial F_x}{\partial j_x}\right)/\rho \tag{57}$$

$$\frac{\partial G_{\mathrm{fluid}}}{\partial j_y} = 9\nu j_y\left(4s_e^2 + s_\nu^2\right)\left(j_x^2 + j_y^2\right)/\rho^3 - 3\nu\left[4s_e^2 e j_y + 3s_\nu^2\left(-j_y p_{xx} + 2j_x p_{xy}\right)\right]/\rho^2 - \left(F_y + 24\nu s_e^2 j_y + j_y\frac{\partial F_y}{\partial j_y}\right)/\rho \tag{58}$$

$$\frac{\partial G_{\mathrm{fluid}}}{\partial p_{xx}} = 9\nu s_\nu^2\left(-j_x^2 + j_y^2 + p_{xx}\rho\right)/\left(2\rho^2\right) \tag{59}$$

$$\frac{\partial G_{\mathrm{fluid}}}{\partial p_{xy}} = 18\nu s_\nu^2\left(-j_x j_y + p_{xy}\rho\right)/\rho^2. \tag{60}$$

The derivatives with respect to the remaining moments all go to zero. With these derivatives the adjoint collision equation (Eq. (34)) is adapted to the specific optimization problem. The adjoint moments are given as:

$$\boldsymbol{n} = \left[\tilde{\rho}, \tilde{e}, \tilde{\epsilon}, \tilde{j}_x, \tilde{q}_x, \tilde{j}_y, \tilde{q}_y, \tilde{p}_{xx}, \tilde{p}_{xy}\right]^{\mathrm{T}}. \tag{61}$$

The remaining steps of transforming the adjoint moments to adjoint distributions (Eq. (36)), streaming the adjoint distribution (Eq. (42)), and transforming the adjoint distributions back to adjoint moments (Eq. (38)) are independent of the optimization objective. The local sensitivity (Eq. (46)) evaluates to:

$$\frac{\mathrm{D}J}{\mathrm{D}\gamma(\boldsymbol{x})} = \frac{\mathrm{d}\alpha(\boldsymbol{x})}{\mathrm{d}\gamma(\boldsymbol{x})} \sum_{t=t_0}^{t_1} \left[\frac{j_x^2 + j_y^2}{\rho} + \left(\frac{\partial \boldsymbol{C}}{\partial \alpha(\boldsymbol{x})} \right)^{\mathrm{T}} \cdot \boldsymbol{n}(\boldsymbol{x}) \right]. \tag{62}$$

For steady problems the sensitivity is computed only for the final state of the adjoint problem. No summation over time is required in this case. In order to apply our approach to other optimization problems not related to dissipation minimization with volume constraint, the cost function and the corresponding adjoint collision term in Eq. (34) has to be derived again. The cost function and its derivatives in the adjoint collision always need to be adjusted to particular optimization problem at hand.

6. Validation and application

The approach presented in the last section is applied to the topology optimization of steady and weakly compressible flow problems. The design variables of the examples are updated by the method of moving asymptotes (MMA [12]). MMA is an iterative strategy used to solve inequality-constrained optimization problems. In each optimizing step of the MMA, the primal problem is approximated by a sub-problem, which ensures the original gradients and constraints. All the sub-problems are solved by a primal–dual interior point method. By adjusting the parameters in each optimizing step, MMA finally gives the converged solution of the optimization problem. A volume constraint is put into the optimization strategy. It is expressed as follows:

$$r = \sum_{\boldsymbol{x}} \gamma(\boldsymbol{x}) - \beta \leq 0 \tag{63}$$

where β is the upper limit of the fluid volume fraction. A volume constraint is required to avoid the undesired optima of all cells being either fluid or solid. The derivative of the volume constraint residual is:

$$\frac{\mathrm{d}r}{\mathrm{d}\gamma(\boldsymbol{x})} \equiv 1. \tag{64}$$

In order to balance the magnitude of the value of the objective and the volume constraint, the objective and the constraint should be normalized using the value of the objective and the constraint residual of the first optimizing step respectively.

The iterative optimization proceeds as depicted in Fig. 4.

In this section, we provide several examples to validate our approach. The first three examples are classic ones commonly found in the fluid mechanics literature. They are used here to give a demonstration and validation of our methodology. They are compared to results of Borrvall et al. [3], which are based on second order numerical solutions of Stokes flow. The diffuser example is presented to demonstrate that our approach is reasonable. The second example considers double pipes in long and short domains. This example shows that the sensitivity can lead to different topologies when the initial domain is just slightly changed. Thirdly, we give results for bending pipes when the inlet and outlet are at a 90 degrees angle. These results demonstrate that our approach works for a wide range of Reynolds numbers. Lastly, we demonstrate our method for the optimization of bifurcating channels. This test case is implemented with different initial setups of the volume limit and the Reynolds number and it demonstrates how the volume constraints affect the final layout shape in our approach.

To reduce the influence of the boundary condition on the optimization process we provided guiding pipes from inlet and outlet boundaries to the device. These pipes are excluded from the optimization domain. The real-world devices would be connected by pipes towards the outside world and the flow in the device would usually influence the flow in these pipes. It is therefore more reasonable to add guiding pipes than to put fixed boundary conditions right to the edges of the design domain. In this paper all examples but two are analysed with guiding pipes attached to both inlets and outlets. Similar considerations have been applied to many examples [40] and are regarded as canonical. We are comparing our results to results of Borrvall et al. [3] where no guiding pipes were used. One test is found to be sensitive to the existence of guiding pipes and we provide an optimization without guiding pipes to show that we can reproduce Borrvall's results with equivalent boundary conditions. Since we study low Reynolds number examples we use relatively short guiding pipes which are as wide as long. Using longer pipes has no influence on the results in the low Reynolds number range.

6.1. 2D diffuser

A two dimensional diffuser example is subsequently presented as a validation test case. The design area is a square domain with left and right sides open. The objective is minimal dissipation with volume fraction 0.5. The shape of the non-optimized domain is shown in Fig. 5(a), where an outlet guiding pipe is hidden. The purpose of the diffuser is to guide a developed flow with parabolic velocity profile into a smaller outlet. Therefore, the developed inflow profile is specified at the left side of design domain without guiding pipe at the inlet. The Reynolds number is set to one since the example is

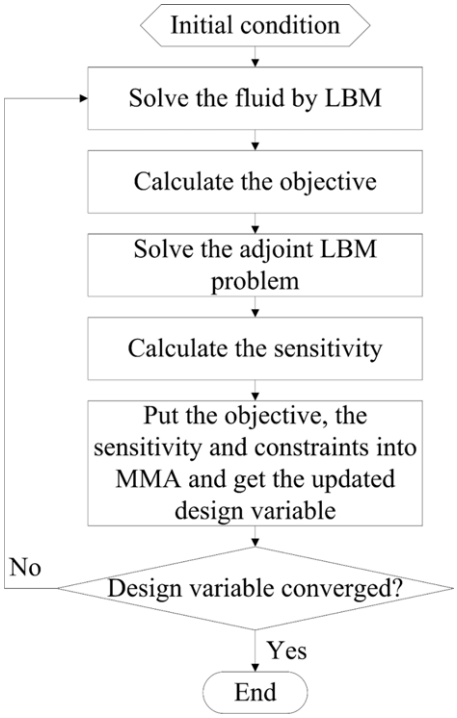


Fig. 4. Flowchart of the optimizing procedure.

being compared with the same Stokes flow case analysed by Borrvall and Petersson in [3]. The lattice kinematic viscosity ν is set to 0.1 in most applications in this paper, except for some high Reynolds number cases in the bending pipe validation. Hence, it is straightforward to calculate the maximum inlet lattice velocity by $u_{\max} = \nu \mathrm{Re}/NL$ where NL is the number of lattice nodes at the inlet. The real fluid medium is water for all examples in this paper. The physical characteristic length in this case (i.e. l in the map) is set as 1 mm. Thus all physical variables can be calculated. In physical units the parameter $\alpha_{\max}$ is set to 3000 s^{-1} and does not depend on the resolution. Results for different resolutions are presented in Fig. 5(b) and (c). They took 320 and 400 optimization steps, respectively, to reach a converged state as presented. There are several criteria to detect the convergence of the topology like checking the change rate of the objective and checking that of the topology. Here we used visual inspection of the design to terminate the process. The contour line of Borrvall's result (100×100 grid points) was digitally extracted from the figure presented in [3] and is compared to our result (99×99 lattice) in Fig. 5(d).

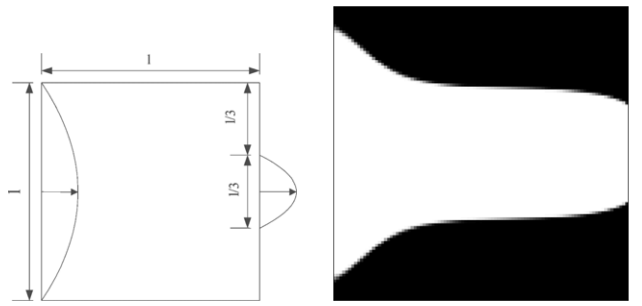
In this example we also evaluate the effect of the choice of the parameters $\alpha_{\max}$. To compare with the original parameter, another coarse meshed (99×99 lattice) optimization is done, in which $\alpha_{\max}$ is 500 s^{-1} and all other parameters remain the same. For simple and convex examples like the diffuser, the final topologies are independent of this small change, but the objective convergence paths are totally different. In the $\alpha_{\max} = 500\ \mathrm{s}^{-1}$ case the topology converges very fast within 15 steps (Fig. 6). However from the velocity field we notice that the permeation in the porous media is strong. On the other hand, when we use the original parameter (Fig. 6), it takes 40 steps to get a clear and reasonable shape and another 280 steps to obtain a better layout (i.e. the design variables γ in the solid areas are mostly less than 0.01). However, this pays off since nearly no permeation can be found. In this paper we focus on the correctness of our approach rather than on efficiency and chose the parameters such that permeation is minimal even if this increases the required number of optimization cycles.

Since our code used MATLAB the computational performance was pretty low at 2.54 million lattice node updates per second for the forward problem and 2.46 million lattice node updates per second for the adjoint problem at a resolution of 99×99 nodes. These performance results are very preliminary. Still it is seen that the forward and adjoint computations have very similar cost.

6.2. Double pipe

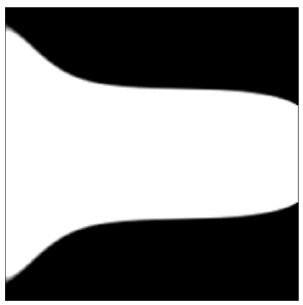
The double pipe test is another example first introduced by Borrvall et al. [3]. The shape of the design domain is shown in Fig. 7(a). The physical characteristic length $l/6$ is 1 mm. This example shows that the method is able to select the optimal topology when there are multiple possibilities. In the longer domain dissipation is minimized by a single pipe, while in a short domain two pipes are optimal.

In this paper, the Reynolds number is set to 1 and the volume fraction is one third in order to present the same setup as used by Borrvall. We apply $\alpha_{\max} = 800\ \mathrm{s}^{-1}$ to both geometries, which have lattice sets of 96×96 and 144×96 grid points,

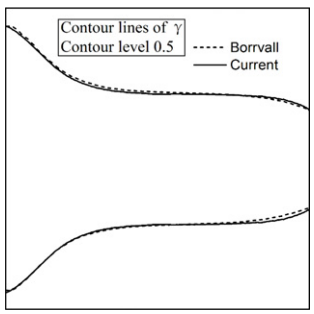


(a) Design domain (l is a size unit).

(b) 99 × 99 lattice result.



(c) 198 × 198 lattice result.



(d) Comparison (coarse lattice).

Fig. 5. 2D diffuser example: design domain (a) and results at different resolutions (b) and (c). It is seen that we obtain a grid independent result even at a moderate resolution. Comparison (d) shows the difference between our result for the diffuser to the result obtained by Borrvall et al. [3].

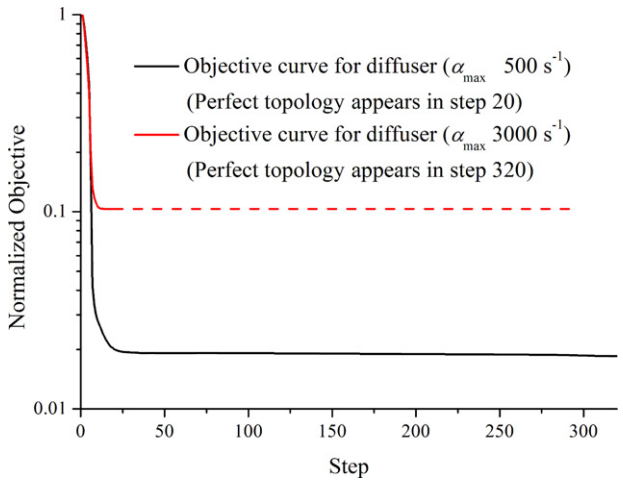
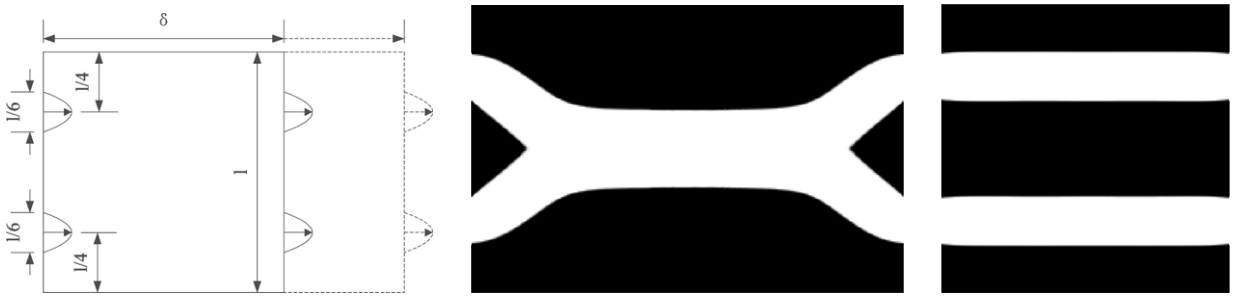


Fig. 6. Comparison of two different α_{max} for the diffuser example (99 × 99 lattice).

respectively. The objective convergence curve is shown in Fig. 8 to make the evolution clear. The figure clearly reveals where the two topologies diverge and what destinations they are targeting. In the $\delta = 1.5l$ case there are a few sudden increases of the cost function after iteration 100. These increases are caused by determining whether the decreasing of the objective or the volume constraint takes the lead. The volume fraction is altered from 0.85 to its final value during the first 25 iterations to facilitate the localization of the optimal topology. This avoids the process from dropping into a local minimum in the first few iterations. After this initial period the optimal topology has been detected but more iterations are required to obtain a converged optimal design. Hence, the total number of optimizing steps for the short domain is 600 and that for the long domain is 1200. The final results are shown in Fig. 7(b) and (c). The results of Borrvall et al. (100 × 100 grid for the short domain case and 150 × 100 for the long domain case) were again extracted from the corresponding figure in [3] and are presented together with the results obtained in this work in Fig. 9(a) and (b).



(a) Design domain (l is a size unit). (b) $\delta = 1.5l$ result. (c) $\delta = l$ result.

Fig. 7. Double pipe example: design domain (a) and the results for a long (b) and short (c) domain respectively.

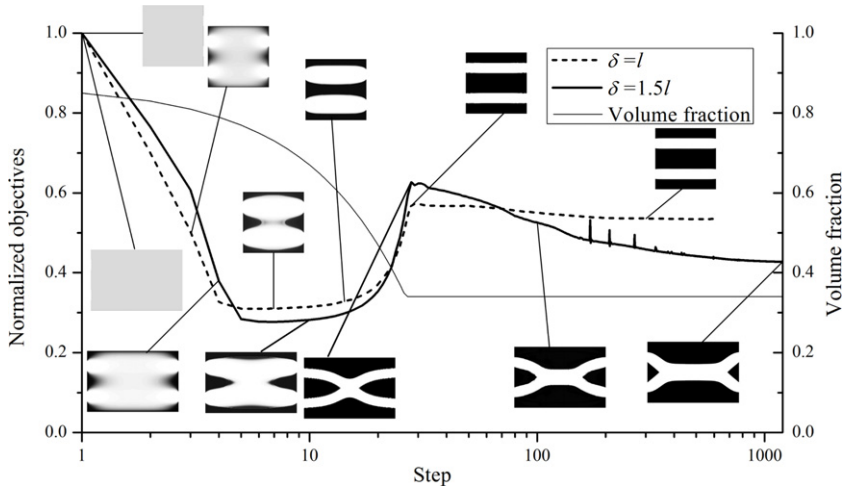
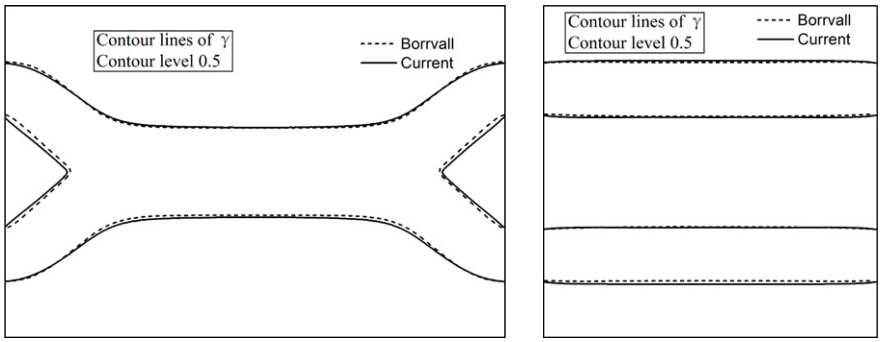


Fig. 8. Double pipe objective curve. The volume fraction is reduced linearly during the first 25 optimization steps (note the logarithmic scale of the step-axis). During the change of the volume fraction the objective can grow. The optimal topology is determined in the first few iterations. The final shape is only found later in the process after the volume fraction reached its final value.



(a) $\delta = 1.5l$ result. (b) $\delta = l$ result.

Fig. 9. Comparison of our results to the ones obtained in [3]. The contour for $\gamma = 0.5$ is shown for the current simulation. The result of Borrvall et al. was extracted digitally from their paper.

6.3. Bending pipe

As a last comparison to the results of [3] we consider a bending pipe example. The design domain is depicted in Fig. 10(a). Borrvall et al.'s result is again extracted as contour lines and presented together with ours in Fig. 11. The volume fraction is 0.255 (Borrvall et al. set it to 0.08π). We present comparable results in both coarse and fine meshes (Fig. 10(b) and (c) and relevant parameters are shown in the first column of Table 1). Our initial result (coarse mesh) does not match Borrvall et al.'s result (100×100 grid) very well. The reason is that the shape of the bending pipe is very sensitive to the existence of guiding pipes at the inlet and the outlet, and that Borrvall et al. did not use guiding pipes. They fixed a parabolic inflow

(a) Design domain (l is a size unit). (b) Re = 1, 100 × 100 lattice. (c) Re = 1, 200 × 200 lattice.

Fig. 10. Bending pipe: design domain (a) and results at different resolutions (b) and (c) assuring grid independence.

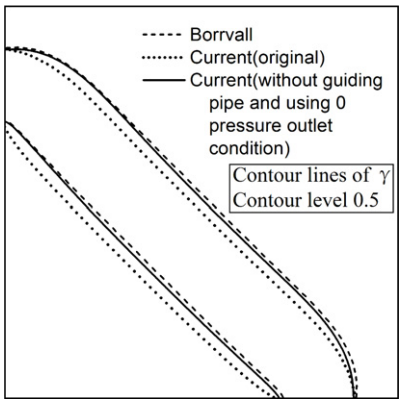


Fig. 11. Comparison of our bending pipe results to the ones obtained in [3]. Borrvall et al. did not use guiding pipes at the inlet and the outlet which explains the difference to our result when guiding pipes are used.

Table 1
Parameters for different Reynolds numbers in the bending pipe example. To avoid unsteady flow, the topology of the Reynolds number 400 case is initialized as a quarter of a ring. In the other cases the design variables are initialized homogeneously. In the Reynolds number 100 case the volume fraction was decreased until it reached the final limit. All other cases were run with constant volume fraction.

Reynolds number	1	10	100	400[a]
Fluid	Water	Water	Water	Water
Characteristic length ($l/5$) (m)	0.001	0.001	0.01	0.01
Maximum inlet velocity (m/s)	0.001	0.01	0.01	0.04
Lattice viscosity (ν)	0.1	0.1	0.01	0.01
α_{max} (s^{-1})	1000	800	8	5
Initial volume fraction	0.255	0.255	0.8[b]	0.255
Final volume fraction	0.255	0.255	0.255	0.255
Lattice	200 × 200	300 × 300	400 × 400	600 × 600
Total optimizing step	400	500	750	800

[a] Initialized as a quarter of a ring.
[b] Volume fraction decreases in the rate of 0.02 per step until it reaches the final limit.

boundary condition and a zero pressure outflow boundary condition directly onto the design domain. To compare with their results, we repeat the optimization for Re = 1 using similar boundary conditions to those in [3]. For the outlet we could use any zero pressure boundary scheme. We chose the one suggested by Zou and He [41]. The current approach is seen to compare favourably to those of Borrvall et al. if equivalent boundary conditions are used. This example highlights the role of the guiding pipes for the optimization result (Fig. 11).

In all considered examples we could reproduce the results Borrvall et al. provided if only we used equivalent boundary conditions. The approach of Borrvall et al., however, is restricted to very low Reynolds numbers since it is based on the Stokes equation. Our approach, just like any other Navier–Stokes equation or LBM based approaches referred in the introduction, has no such restrictions. We demonstrate in Fig. 12 that only the low Reynolds (Stokes) case has a straight line as optimum. At increasing Reynolds numbers the optimal pipe gets increasingly bent. Relevant parameters are shown in Table 1.

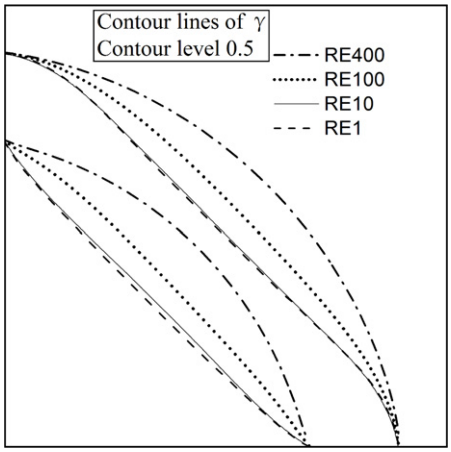


Fig. 12. Bending pipe results obtained for different Reynolds numbers. A straight pipe is only obtained in the limit of Stokes flow. At increasing Reynolds numbers the increasing curvature leads to lower dissipation.

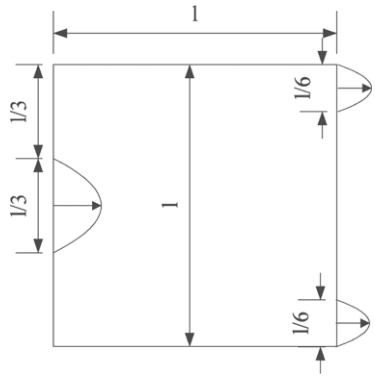


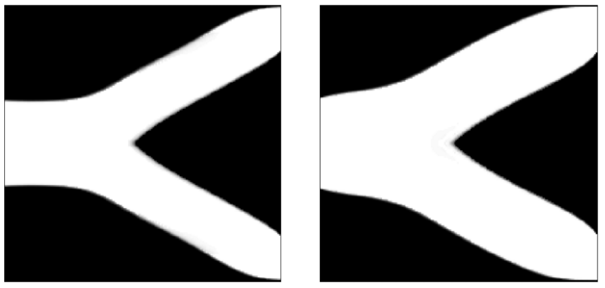
Fig. 13. Design domain (setup) of the bifurcating channels example (l is a size unit).

In the Reynolds number 100 case, the decreasing volume upper limit is used again. The current case has no wrong local optimum but decreasing the volume limit is still beneficial for the efficiency as it avoids the flatness of the sensitivity due to the loss of degrees of freedom imposed by a lower volume limit. Compared to the fixed post-normalized volume constraint gradient (Eq. (64)), the normalized sensitivity could also become too weak to keep the robustness of the MMA algorithm after a number of optimizing steps if decreasing the volume upper limit was not applied. In the Reynolds number 400 case, a common initial condition of homogeneous material density is not applied. On the contrary a formed shape is set before the optimization starts. This kind of implementation is thought to be unconvincing since the final topology is very close to the initial setup. However, if a homogenized initial topology is used, the flow would be unsteady at the first step. Under this circumstance the flow field would not converge to a steady state and optimization could not continue. In this example, a quarter of a ring is selected as the initial topology. Other topologies which do not cause unsteady flows (e.g. the result of the Reynolds number 100 case, which in our test converges to a result exactly the same as starting from a quarter of a ring) could also work. This indicates that for such convex problems the optima do not rely on the initial conditions. Similar bending pipe examples for high Reynolds number flows have been proposed by Gersborg-Hansen [42] based on the Navier–Stokes equation.

6.4. Bifurcating channels

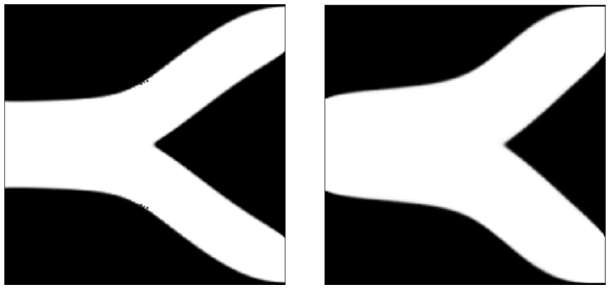
Our last example is the optimization of bifurcating channels. The domain is a square as can be seen in Fig. 13. The layout topology is optimized for two volume fractions and two Reynolds numbers. The physical characteristic length $l/3$ is 1 mm, and we use 192 $\times$ 192 lattice points. For the Reynolds number 10 cases $\alpha_{\max}$ is 2000 s^{-1} and that for the Reynolds number 50 cases is 1000 s^{-1}. Results are presented in Fig. 14.

In this example decreasing the volume is again applied. The reason is very similar to that in the Reynolds number 100 bending pipe example. We use the Reynolds number 50 and volume fraction 0.5 bifurcating channels to analyse the influence of decreasing the volume fraction during the optimization. The results obtained with and without decreasing volume are presented in Fig. 15.



(a) Reynolds number 10, volume fraction 0.4.

(b) Reynolds number 10, volume fraction 0.5.



(c) Reynolds number 50, volume fraction 0.4.

(d) Reynolds number 50, volume fraction 0.5.

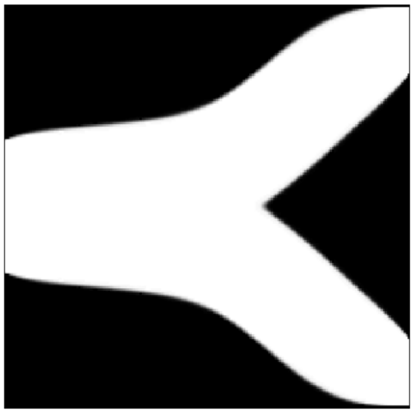
Fig. 14. Optimized layouts of bifurcating channels for two volume fractions and two Reynolds numbers.

In Fig. 15(a), the topology is obtained through decreasing the volume fraction by 0.01 per step starting from 1. The figure also reveals that the channels are clear and without obstacles. From Fig. 15(b) it is also acquainted that the objective curve is very smooth. However, in Fig. 15(c) the same test case is optimized with a fixed volume fraction. The topology appears to be worse than in Fig. 15(a). There are several grey areas in the channel, which apparently increase the energy dissipation. They are not easily visible in Fig. 15(c) but we provide the same plot with the darkness of the grey areas artificially enhanced in Fig. 16. The convergence curve of this case is also unstable (Fig. 15(d)). The comparison demonstrates that a changing volume makes the optimization problem more flexible and the sensitivity stronger. It can also be seen that decreasing the volume fraction increases the efficiency of the optimization process. Without decreasing the volume fraction the task needs 600 steps to get a result that is still not good, while the optimization for all other cases (i.e. Reynolds number 10 and 50, final volume fraction 0.4 and 0.5) finishes within 300 steps and reaches a lower absolute objective. The application of this trick does not make a big difference when solving the Reynolds number 10 problem.

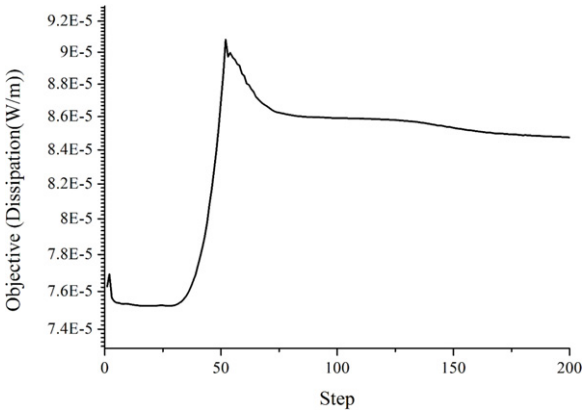
Comparing Fig. 14(a) to Fig. 14(c), we can see that the acute angle between the two branches of the bifurcating channels increases for larger Reynolds numbers. This phenomenon becomes clearer when comparing Fig. 14(b) to Fig. 14(d) which show wider channels simply because they are allowed to have a larger volume fraction than Fig. 14(a) and (c). In general, a larger volume will result in less dissipation for low Reynolds number flow.

7. Conclusions

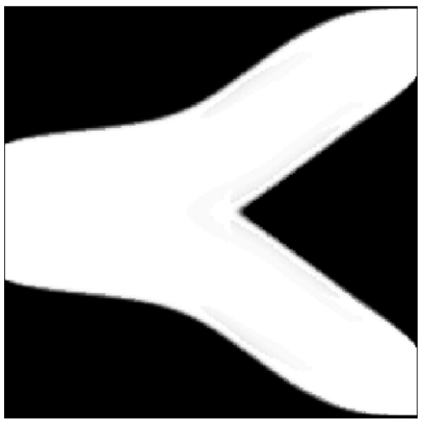
In this paper we presented a general methodology for fluidic topology optimization using the multiple-relaxation-time lattice Boltzmann equation. The adjoint lattice Boltzmann equation was explicitly derived and the procedure was explained in detail. Taking the adjoint of the lattice Boltzmann method in moment space led to a simplification of the derivation compared to approaches based on the distribution function (i.e. [24,17]) and provides more insight than automatic differentiation of the lattice Boltzmann code (i.e. [43]). Our approach recycles the numerical differentiation from the forward problem at low computational cost by extracting the stress tensor from the moments. Hence, while the considered cost function depends on the gradients of the velocity these gradients did not need to be computed explicitly. The algorithm used for the adjoint lattice Boltzmann method is essentially the same as the algorithm for the usual lattice Boltzmann algorithm. The method can be easily incorporated into any existing lattice Boltzmann code. All characteristic features of the lattice Boltzmann method, such as efficient parallelization, are also features of the adjoint lattice Boltzmann method. We demonstrated the viability of our approach by detailed comparisons to results obtained for the Stokes equation. Our approach produced virtually identical results to those obtained by Borrvall et al. [3]. Unlike the approach based on the Stokes equation our method, as well as many other Navier–Stokes equation or LBM based methods, can in principle be used for a wide range of Reynolds numbers which we demonstrated with a bending pipe test case. All presented examples considered



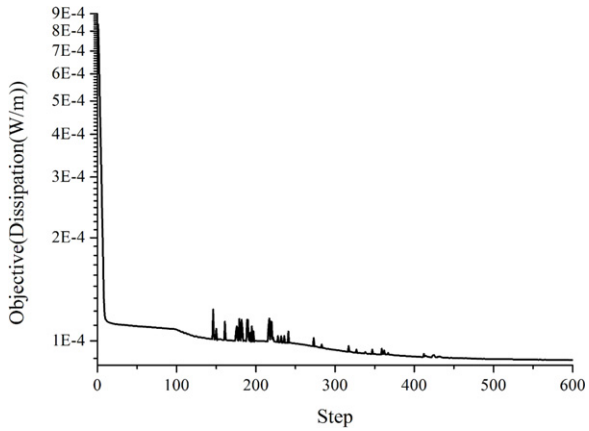
(a) Result with decreasing volume fraction.



(b) Objective curve for optimization with decreasing volume fraction. (Re = 50, Volume fraction = 0.5, Final Objective = 8.4745e−5 W/m).



(c) Result without decreasing volume fraction.



(d) Objective curve for optimization without decreasing volume fraction. (Re = 50, Volume fraction = 0.5, Final Objective = 8.7999e − 5 W/m).

Fig. 15. Comparison of the topology results of Re = 50 and 0.5 volume fraction bifurcating channels with and without decreasing the volume fraction during optimization. The objective on the left hand side is given in absolute values in order to make the two plots comparable. Decreasing the volume fraction leads to an increasing objective in the beginning while starting from the final volume fraction leads to an immediate reduction of the objective during the first iterations. The absolute value of the final objective in (b) is lower than in (d).

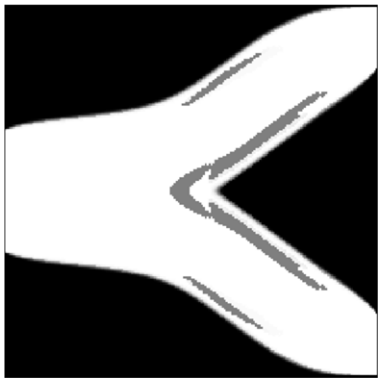


Fig. 16. The same optimization result as in Fig. 15(c) with the grey areas in the channel artificially enhanced in darkness to improve visibility. While these islands are not easily recognised without the enhancement they still reduce the quality of the optimization result. The formation of islands can be avoided by changing the volume fraction during the early stage of the optimization as done in Fig. 15(a).

steady flows while the procedure is equally valid for unsteady flows. The difficulties associated with the optimization of unsteady flows are not of theoretical nature. The adjoint sensitivity analysis requires the complete time history of the forward problem. Hence, any practically relevant time dependent optimization problem would be extremely expensive

in terms of its memory requirements even if compression techniques are used. In the case of steady state problems the procedure does not require a lot of memory and comes at a similar computational expense as the forward problem. Steady state topology optimization of fluidic devices of engineering relevance is perfectly feasible with this method.

Acknowledgements

This work is supported by National Science Funds of China (No. 50975272 and No. 51275504) and the Science and Technology Development Plan of Jilin Province, China (No. 20140519007JH). And the authors are grateful to Prof. Français Dubois for his valuable advice on the theoretical part of this paper. Geng Liu also wants to thank Dr. Mathias Krause for his advices and discussions on implementation. We thank Andrea Pasquali for proofreading the manuscript.

References

[1] M.P. Bendsøe, O. Sigmund, Material interpolation schemes in topology optimization, Arch. Appl. Mech. (Ingr. Arch.) 69 (9–10) (1999) 635–654.
[2] G.I.N. Rozvany, Aims, scope, methods, history and unifiedterminology of computer-aided topology optimization in structural mechanics, Struct. Multidiscip. Optim. 21 (2) (2001) 90–108.
[3] T. Borrvall, J. Petersson, Topology optimization of fluids in Stokes flow, Internat. J. Numer. Methods Fluids 41 (1) (2003) 77–107.
[4] A. Gersborg-Hansen, O. Sigmund, R.B. Haber, Topology optimization of channel flow problems, Struct. Multidiscip. Optim. 29 (2005) 1–12.
[5] Y. Deng, Z. Liu, P. Zhang, Y. Wu, Topology optimization of unsteady incompressible Navier–Stokes flow, J. Comput. Phys. 230 (2011) 6688–6708.
[6] M.Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, Comput. Methods Appl. Mech. Engrg. 192 (1–2) (2003) 227–246.
[7] X.-B. Duan, Y.-C. Ma, R. Zhang, Shape-topology optimization of Stokes flow via variational level set method, Appl. Math. Comput. 202 (1) (2008) 200–209.
[8] Z. Liu, J.G. Korvink, Adaptive moving mesh level set method for structure topology optimization, Eng. Optim. 40 (6) (2008) 529–558.
[9] S. Zhou, Q. Li, A variational level set method for the topology optimization of steady-state Navier–Stokes flow, J. Comput. Phys. 227 (24) (2008) 10178–10195.
[10] V.J. Challis, J.K. Guest, Level set topology optimization of fluids in Stokes flow, Internat. J. Numer. Methods Engrg. 79 (10) (2009) 1284–1308.
[11] S. Kreissl, K. Maute, Levelset based fluid topology optimization using the extended finite element method, Struct. Multidiscip. Optim. 46 (3) (2012) 311–326.
[12] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, Internat. J. Numer. Methods Engrg. 24 (2) (1987) 359–373.
[13] H.W. Kuhn, A.W. Tucker, Nonlinear programming, in: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, Calif., 1951, pp. 481–492. URL: http://projecteuclid.org/euclid.bsmsp/1200500249.
[14] Y. Deng, P. Zhang, Y. Liu, Y. Wu, Z. Liu, Optimization of unsteady incompressible Navier–Stokes flows using variational level set method, Internat. J. Numer. Methods Fluids 71 (12) (2013) 1475–1493.
[15] S. Kreissl, G. Pingen, K. Maute, Topology optimization for unsteady flow, Internat. J. Numer. Methods Engrg. (2011) 1229–1253. http://dx.doi.org/10.1002/nme.3151.
[16] M.M. Tekitek, M. Bouzidi, F. Dubois, P. Lallemand, Adjoint lattice Boltzmann equation for parameter identification, Comput. & Fluids 35 (8–9) (2006) 805–813.
[17] G. Pingen, A. Evgrafov, K. Maute, Topology optimization of flow domains using the lattice Boltzmann method, Struct. Multidiscip. Optim. 34 (6) (2007) 507–524.
[18] G. Pingen, A. Evgrafov, K. Maute, 3D topology optimization of fluids by the lattice Boltzmann method, in: 11th AIAAISSMO Multidisciplinary Analysis and Optimization Conference, Vol. 4, 2006, pp. 1–13.
[19] G. Pingen, 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.
[20] G. Pingen, M. Waidmann, A. Evgrafov, K. Maute, A parametric level-set approach for topology optimization of flow domains, Struct. Multidiscip. Optim. 41 (1) (2009) 117–131.
[21] S. Kreissl, G. Pingen, A. Evgrafov, K. Maute, Topology optimization of flexible micro-fluidic devices, Struct. Multidiscip. Optim. 42 (4) (2010) 495–516.
[22] D. Makhija, G. Pingen, R. Yang, K. Maute, Topology optimization of multi-component flows using a multi-relaxation time lattice Boltzmann method, Comput. & Fluids 67 (2012) 104–114.
[23] A. Kirk, S. Kreissl, G. Pingen, K. Maute, Lattice Boltzmann topology optimization for transient flow, in: MAESC 2011 Conference, Christian Brothers University, Memphis, Tennessee, 2011.
[24] M.J. Krause, G. Thäter, V. Heuveline, Adjoint-based fluid flow control and optimisation with lattice Boltzmann methods, Comput. Math. Appl. 65 (6) (2012) 945–960.
[25] U. Frisch, D. d'Humieres, B. Hasslacher, P. Lallemand, Y. Pomeau, J.P. Rivet, Lattice gas hydrodynamics in two and three dimensions, Complex Syst. 1 (4) (1987) 649–707.
[26] M. Junk, A. Klar, L.-S. Luo, Asymptotic analysis of the lattice Boltzmann equation, J. Comput. Phys. 210 (2) (2005) 676–704. http://dx.doi.org/10.1016/j.jcp.2005.05.003.
[27] F. Dubois, Equivalent partial differential equations of a lattice Boltzmann scheme, Comput. Math. Appl. 55 (7) (2008) 1441–1449.
[28] L.-S. Luo, W. Liao, X. Chen, Y. Peng, W. Zhang, Numerics of the lattice Boltzmann method: effects of collision models on the lattice Boltzmann simulations, Phys. Rev. E 83 (2011) 056710. http://dx.doi.org/10.1103/PhysRevE.83.056710. URL: http://link.aps.org/doi/10.1103/PhysRevE.83.056710.
[29] S. Geller, M. Krafczyk, J. Tölke, S. Turek, J. Hron, Benchmark computations based on lattice-Boltzmann, finite element and finite volume methods for laminar flows, Comput. & Fluids 35 (8–9) (2006) 888–897.
[30] E. Özkaya, N.R. Gauger, Automatic transition from simulation to one-shot shape optimization with Navier–Stokes equations, Ges. Angew. Math. Mech. 33 (2) (2010) 133–147.
[31] A. Griewank, Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation, 1991.
[32] Q. Wang, P. Moin, G. Iaccarino, Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation, SIAM J. Sci. Comput. 31 (4) (2009) 2549–2567.
[33] E. Vergnault, O. Malaspinas, P. Sagaut, A time-reversal lattice Boltzmann method, J. Comput. Phys. 230 (2011) 8155–8167.
[34] Y.H. Qian, D. d'Humieres, P. Lallemand, Lattice BGK models for Navier–Stokes equation, Europhys. Lett. 17 (6) (1992) 479–484.
[35] P. Lallemand, L.-S. Luo, Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, galilean invariance, and stability, Phys. Rev. E 61 (6 Pt A) (2000) 6546–6562.
[36] A. Ladd, R. Verberg, Lattice-Boltzmann simulations of particle–fluid suspensions, J. Stat. Phys. 104 (2001) 1191–1251.
[37] M. Krafczyk, J. Tölke, L.-S. Luo, Large-eddy simulations with a multiple-relaxation-time LBE model, Internat. J. Modern Phys. B 17 (01–02) (2003) 33–39.
[38] W.-A. Yong, L.-S. Luo, Accuracy of viscous stress in the lattice Boltzmann equation with simple boundary conditions, Phys. Rev. E (3) 86 (6 Pt 2) (2012) 065701.
[39] M. Stolpe, K. Svanberg, An alternative interpolation scheme for minimum compliance topology optimization, Struct. Multidiscip. Optim. 22 (2001) 116–124.

[40] L.H. Olesen, F. Okkels, H. Bruus, A high-level programming-language implementation of topology optimization applied to steady-state Navier–Stokes flow, Internat. J. Numer. Methods Engrg. 65 (7) (2006) 975–1001. http://dx.doi.org/10.1002/nme.1468.
[41] Q. Zou, X. He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, Phys. Fluids 9 (6) (1997) 1591–1598.
[42] A. Gersborg-Hansen, O. Sigmund, R. Haber, Topology optimization of channel flow problems, Struct. Multidiscip. Optim. 29 (2005) 1–12. http://dx.doi.org/10.1007/s00158-004-0508-7.
[43] M.J. Krause, V. Heuveline, Parallel fluid flow control and optimisation with lattice Boltzmann methods and automatic differentiation, Comput. & Fluids 80 (2013) 28–36.