Optimization of a Micromixer with Automatic Differentiation

Julius Jeßberger [1,2,3,*] **, Jan E. Marquardt** [1,3] **, Luca Heim** [1,3] **, Jakob Mangold** [1,2] **, Fedor Bukreev** [1,3] **and Mathias J. Krause** [1,2,3]

1 Lattice Boltzmann Research Group, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany; jan.marquardt@kit.edu (J.E.M.); luca.heim@gmx.de (L.H.); jhm305@yahoo.de (J.M.); fedor.bukreev@kit.edu (F.B.); mathias.krause@kit.edu (M.J.K.)
2 Institute for Applied and Numerical Mathematics (IANM), Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany
3 Institute for Mechanical Process Engineering and Mechanics (MVM), Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany
* Correspondence: julius.jessberger@kit.edu

Abstract: As micromixers offer the cheap and simple mixing of fluids and suspensions, they have become a key device in microfluidics. Their mixing performance can be significantly increased by periodically varying the inlet pressure, which leads to a non-static flow and improved mixing process. In this work, a micromixer with a T-junction and a meandering channel is considered. A periodic pulse function for the inlet pressure is numerically optimized with regard to frequency, amplitude and shape. Thereunto, fluid flow and adsorptive concentration are simulated three-dimensionally with a lattice Boltzmann method (LBM) in OpenLB. Its implementation is then combined with forward automatic differentiation (AD), which allows for the generic application of fast gradient-based optimization schemes. The mixing quality is shown to be increased by 21.4% in comparison to the static, passive regime. Methodically, the results confirm the suitability of the combination of LBM and AD to solve process-scale optimization problems and the improved accuracy of AD over difference quotient approaches in this context.

Keywords: lattice Boltzmann method; computational fluid dynamics; micromixer; optimization; automatic differentiation

1. Introduction

In micromixers, different fluid components are typically inserted through two different inlets into one curved pipe. Mainly because of their simplicity, they have become a key device in microfluidics [1]. The mixing is then prolonged mostly by advection and also by diffusion. Since the channel geometry has a decisive effect on the overall efficiency of the mixer, many different shapes and versions of this device have been suggested, studied and compared [1–3], e.g., it has been observed that applying non-constant, pulsating external energy sources (active mixing) can significantly increase the mixing efficiency [1]. In [4], Glasgow and Aubry suggested pulsing the inlet flow rates periodically in time, which improved the simulated mixing quality by factors of up to 3.9, measured similarly to Danckwerts' concept of mixing quality [5] for Péclet number $Pe = 3000$.

The many possible design options yield a need for numerical simulations and, even more importantly, systematic optimization, in order to achieve the best possible results. The mixing process inside passive micromixers has been researched via CFD simulations by [6–8], among others. Rudyak and Minakov [9] conducted a fluid dynamic investigation of mixers for a wide range of Reynolds numbers, with periodically varying inlet velocities and a manual selection of the optimal pulse. They showed that the optimal frequency seems to depend on the amplitude and that there is a strong non-linear dependence of the total flow behavior on the Reynolds number. Among others, Maier et al. [10] presented a

multi-component model including fluid flow, advection, diffusion and adsorption of Lagrange particles using the lattice Boltzmann method (LBM), and validated their simulations by experiments.

Santana et al. [11] and S. Hossain et al. [12] used numerical optimization to improve (geometrical) design features of static mixers. For this, Hossain et al. used surrogate models based on three-dimensional simulations. Bockelmann et al. studied electrokinetic micromixers with one inlet and pulsating electric potentials [13]. They used finite element method (FEM) simulations and numerical optimization in order to find an optimally pulsating electric potential, which yielded a performance improvement of almost 300%. Since the Péclet numbers in this context are usually very high, which requires a high spatial resolution, they used two-dimensional simulations on a slice though the midplane of the domain. On the one hand, the particulate concentration field looks similar for slices at different heights of a three-dimensional simulation with height-independent input data, which may give reason to omit simulating the vertical component [14]. On the other hand, three-dimensional effects, such as Dean vortices or secondary flow [3], that may have a large impact on mixing can hardly be captured. To the best of the authors' knowledge, no study is available on the numerical optimization of the pulsation of micromixers based on three-dimensional simulations.

The sensitivity of the objective functional with regard to the design variables, which is necessary for any fast gradient-based numerical optimization method, can, in general, be computed via sensitivity-based and adjoint methods [15]. For the first option, the state-of-the-art (at least in the real world application) is the employment of differences quotients. Alternatively, the forward mode of automatic differentiation (AD) has become popular because of its generic approach and a higher accuracy compared to difference quotients. Adjoint equations can be formulated both on a continuous and on a discrete basis, where the latter has a close connection to the reverse mode of AD and therefore a high genericity as an advantage. In the case of only a few design variables, sensitivity-based methods are typically preferred because the implementation is much simpler while the computational effort is not significantly higher [15].

As a result of the rather few design variables, we consider the forward mode of AD in this work. The combination of forward AD and LBM has been reported in [16,17] and it has later been applied, e.g., in [18,19], but, despite its advantages, it has not yet become a standard design tool for real world applications, and a thorough validation of AD in the context of CFD is still rare.

The goal of this work is the numerical optimization of a pulsating inlet pressure with regard to the frequency, amplitude and phase profile, which has not been performed before. Therefore, three-dimensional LBM-simulations of flow field and particulate concentration are performed, coupled with the steepest descent/L-BFGS method for optimization. The required gradients of the goal functional with regard to the optimization variables are computed using forward AD. Both the usage of three-dimensional simulations for optimization and the combination of LBM and forward AD are new in the context of micromixing. Hence, the numerical properties of this novel approach are validated and presented in detail.

The article is structured as follows: Our models and methods are introduced in Section 2. Therefore, we consider the design of the micromixer that we have studied, the physical modeling, LBM, the setup of the optimization problem, AD and some numerical optimization issues. In Section 3, we present particular numerical studies. The simulation of flow and its derivatives with regard to the design variables are explained and validated. Finally, the results of the numerical optimization are presented.

2. Materials and Methods

2.1. Experimental Design

In this work, a micromixer with two inlets Γ_l, Γ_r, a T-junction, a meandering channel Ω and one outlet Γ_{out}, as depicted in Figure 1, was considered, similar to the one that was

studied in [10]. The channel has a cross section of 1.0×10^{-3} m $\times 2.0 \times 10^{-3}$ m and a total length of 2.7×10^{-2} m. Through the left inlet Γ_l, the adsorptive (solved in the carrier fluid) entered. Through the other inlet, the carrier fluid and particles were inserted. Multiple deflections with a 90 degree angle enhanced the mixing quality.
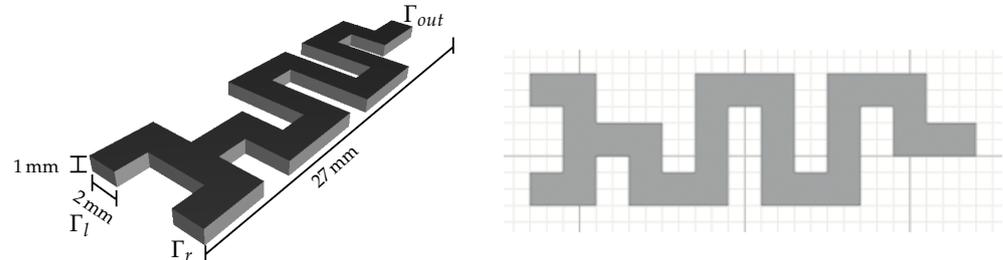
Figure 1. Geometry of the investigated micromixer. (**Left**): Below in the picture are the two inlets Γ_l, Γ_r and the T-junction. On the top, we find the outlet Γ_{out}. (**Right**): Dimensions of the channel in the xy-plane on a grid with width 1 mm.

2.2. Fluid Flow Model

As in [10], we assume that the adsorptive material is dissolved in the carrier fluid. Its concentration is assumed to be significantly higher than the concentration of the particle component. Consequently, a one-way coupled two-phase flow model (carrier fluid + adsorptive) is sufficient in order to evaluate the performance of the mixer, where the fluid dynamics are described by an incompressible Navier–Stokes equation (NSE) and the particulate concentration is modeled in an Euler approach by an advection–diffusion equation (ADE). For a concise introduction into the physical model, we refer to [10,20]. The material constants are listed in Section 3.1.

The laminar fluid flow was driven by a fixed velocity outflow, with pressure boundary conditions at the inlets. Therefore, a Dirichlet fixed-velocity condition with a parabolic profile was set at the outlet, as well as pressure boundary conditions at the two inlets. The pressure at the second inlet varied periodically in time due to the inlet control. At the channel walls, the fluid obtained a Dirichlet no-slip boundary condition.

All of this led to the following system of NSE

$$
\begin{aligned}
\nabla \cdot \boldsymbol{u} &= 0 && \text{in } \Omega, \\
\partial_t \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \nabla p &= \boldsymbol{0} && \text{in } \Omega, \\
\boldsymbol{u} &= \boldsymbol{0} && \text{on } \Gamma_w, \\
\boldsymbol{u}_\tau = \boldsymbol{0}, \quad p &= 0 && \text{on } \Gamma_l, \\
\boldsymbol{u}_\tau = \boldsymbol{0}, \quad p &= \tilde{p}(t) && \text{on } \Gamma_r, \\
\boldsymbol{u}_\tau = \boldsymbol{0}, \quad u_n &= s(t)\tilde{u}(\boldsymbol{x}) && \text{on } \Gamma_{out}, \\
\boldsymbol{u} &= \boldsymbol{0} && \text{for } t = 0
\end{aligned}
\tag{1}
$$

on the time interval $I = [0, T]$, for channel wall Γ_w, velocity $\boldsymbol{u}$ with tangential and normal components $\boldsymbol{u}_\tau$, u_n on the boundary, kinematic viscosity $\nu = 10^{-6}$ m^2/s, kinematic pressure p, a prescribed pressure function $\tilde{p}$ and a rectangular parabolic profile $\tilde{u}$ at the outlet. Lastly, s denotes a start-up coefficient that scales smoothly from 0 to 1. Precisely, we set

$$
s(t) := \begin{cases} \frac{1}{2}[\sin(t\pi/t_1 - \pi/2) + 1] & \text{if } t < t_1, \\ 1 & \text{else} \end{cases}
$$

in order to obtain the start-up interval length $t_1 := 0.6$ s.

The relative mass concentration of adsorptive particles φ was described by an ADE, whereas the concentration φ was fixed to 1 at the inlet as a Dirichlet boundary condition, and a Neumann impermeability condition was set at the second inlet and on the channel walls. At the outlet, free outflow was expected. We obtained

$$
\begin{aligned}
\partial_t \varphi - D\Delta\varphi + \nabla \cdot (\varphi \boldsymbol{u}) &= 0 \quad \text{in } \Omega, \\
\partial_n \varphi &= 0 \quad \text{on } \Gamma_w \cup \Gamma_r \cup \Gamma_{out}, \\
\varphi &= 1 \quad \text{on } \Gamma_l, \\
\varphi &= 0 \quad \text{for } t = 0
\end{aligned}
\tag{2}
$$

on the interval $I = [0, T]$, for concentration φ and diffusion coefficient $D = 10^{-9}$ m^2/s.

2.3. Discretization with a Lattice Boltzmann Method

The fluid flow and the adsorptive concentration were simulated with lattice Boltzmann methods (LBM). These mesoscopic methods have their physical origin in the Boltzmann equation, which describes the evolution of particle distributions in the phase space. By a coupled discretization of space and time, the LBM were obtained, where fluid populations at grid nodes with certain discrete velocities are simulated. This resulted in matrix-free numerical schemes that consisted of collision steps (purely local at each mesh point) and streaming steps (almost local, only neighboring nodes are involved). The macroscopic quantities' density, velocity, pressure, etc., could then be computed as moments of the fluid populations (cf. [21]). Beginning with NSE, many variants have been designed in order to solve, e.g., the ADE, multi-component flows and chemically reacting flows numerically. For the NSE and ADE, the schemes with the BGK collision operator (due to Bhatnagar, Gross and Krook [22]) have been proven to be second-order consistent for diffusive scaling ($\Delta t \sim \Delta x^2 \to 0$) via asymptotic expansion [16,21,23]. Due to their inherent locality, LBM offer good parallelization properties [24], which make them attractive, especially for large-scale simulations. For a concise introduction into LBM, we recommend the book of Krüger et al. [21].

In this work, the LBM with BGK collision operator and D3Q19 stencil was applied for the NSE. For the ADE, the advection diffusion BGK collision [21] was supplemented with a D3Q7 stencil and first-order equilibrium operator, as well as a first-order upwind scheme for stabilization [20,25].

Since particle distributions are simulated instead of the macroscopic variables, the classical macroscopic boundary conditions have to be adapted for LBM in order to prescribe incoming populations at the boundary nodes. In this work, we utilized typical formulations that correspond to the macroscopic conditions as stated in (1) and (2). Thus, we used a fullway bounce-back rule at the channel walls for both NSE and ADE [21] as a homogeneous Dirichlet/Neumann boundary condition, respectively. A fixed velocity boundary condition according to [26] was set for the NSE at the outlet and a pressure boundary condition due to [27] for the NSE was used at the inlets. Discrete in- and outflow conditions for the ADE were set according to [28] and [20], respectively.

2.4. Optimization Setup

In the optimization process, a phase function $\tilde{p}(t)$ in (1) was determined, such that the mixing quality at the outlet is maximized. The discrete versions of the governing Equations (1) and (2) were treated as side conditions [29].

For the periodic part of the phase function $\tilde{p}(t)$, two different approaches were followed: a composite cosine function and a smooth approximation of a square wave. The first one is defined as

$$
p_{\text{coscomp}}(t) := \begin{cases} a\cos\left(\dfrac{2\pi(t \bmod t_p)}{dt_p}\right) & \text{if } t \bmod t_p \leq dt_p, \\[2mm] a\cos\left(\dfrac{2\pi(t \bmod t_p)}{(1-d)t_p}\right) & \text{else,} \end{cases}
\tag{3}
$$

where the free variables amplitude $a \in \mathbb{R}^+$, period length $t_p \in \mathbb{R}^+$ and phase difference $d \in (0, 1)$ have to be optimized (cf. Figure 2) [29]. Via defining the continuous modulo-Operator as $x \bmod y := x - \max_{l \in \mathbb{Z}, ly < x} ly$, it is differentiable in both arguments almost everywhere.
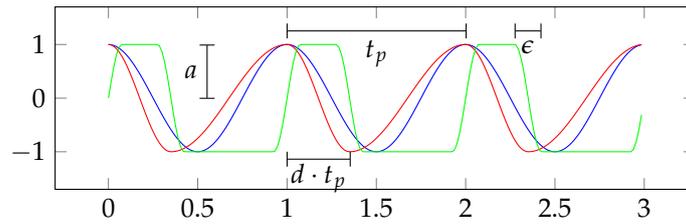
Figure 2. Cosine (blue), composite cosine $p_{\text{coscomp}}(t)$ (red) and smoothed square wave $p_{\text{ssqw}}(t)$ (green, with larger ϵ for visualization).

The smoothed square wave is defined as

$$p_{\text{ssqw}}(t) := \begin{cases} a\sin(\pi(t \bmod t_p)/\epsilon) & \text{if } t \bmod t_p \leq \frac{1}{2}\epsilon, \\ a & \text{if } \frac{1}{2}\epsilon < t \bmod t_p \leq dt_p - \frac{1}{2}\epsilon, \\ -a\sin(\pi(t \bmod t_p) - dt_p/\epsilon) & \text{if } dt_p - \frac{1}{2}\epsilon < t \bmod t_p \leq dt_p + \frac{1}{2}\epsilon, \\ -a & \text{if } dt_p + \frac{1}{2}\epsilon < t \bmod t_p \leq t_p - \frac{1}{2}\epsilon, \\ a\sin(\pi(t \bmod t_p - t_p)/\epsilon) & \text{else,} \end{cases} \qquad (4)$$

with parameters a, t_p, d as above and fixed smoothing interval length $\epsilon := 1.0 \times 10^{-2}$ s (cf. Figure 2). Independent of regularity questions concerning the well-posedness of the NSE (1), smoothing the classical square wave is necessary for a gradient-based optimization with regard to t_p.

In order to achieve a smooth start-up up to time $t_2 := 1.0 \in [0, T]$, we then used

$$\tilde{p}(t) := \begin{cases} 0 & \text{if } t \leq t_1, \\ \frac{a}{2}\cos\left(\frac{2\pi(t-t_1)}{t_2-t_1}\right) - \frac{a}{2} & \text{if } t_1 < t \leq t_2, \\ p_{\text{coscomp}}(t) & \text{else} \end{cases}$$

or a respective version for p_{ssqw} in the NSE (1).

For both pulse functions, the amplitude a needs to be bounded from above so that no backflow happens through the inlets. Technical limitations and how much pressure can be applied at the inlets were not considered in this study.

There are several approaches to evaluate the mixing quality; we followed Danckwerts' concept of segregation intensity [5], where the variance of concentration φ is compared to the variance of the concentration of a fully segregated composition. Since we could expect almost periodic flow, the averaged mixing quality over one time period $[T - t_p, T]$ was considered. We defined average concentration $E(\varphi) := \int_{T-t_p}^{T} \int_{\Gamma_{out}} \varphi(t, x) \, dx \, dt$, variance $\sigma^2(\varphi) := \int_{T-t_p}^{T} \|\varphi(t, \cdot) - E(\varphi)\|_{L^2(\Gamma_{out})}^2 \, dt$ and finally the segregation intensity

$$J(\varphi) := \frac{\sigma^2(\varphi)}{E(\varphi)(1 - E(\varphi))}. \qquad (5)$$

of concentration φ on the time interval $[T - t_p, T]$. The mixing quality could then be defined as

$$\tilde{J}(\varphi) = 1 - \sqrt{J(\varphi)}, \qquad (6)$$

according to [30]. In this work, the segregation intensity was minimized, which is equivalent to a maximization of the mixing quality.

2.5. Automatic Differentiation

The accuracy of numerical derivatives computed with difference quotients cannot be increased arbitrarily by reducing the step size, but, due to rounding errors, it rises if the steps width goes below a certain optimal value [31]. In double precision (accuracy $\sim 10^{-16}$), this optimum is known to be $h \approx 10^{-8}$ for forward (FDQ) and $h \approx 10^{-5}$ for central

difference quotients (CDQ), where, typically, 8 and 6 digits are lost, respectively. This issue can be avoided with AD methods.

With AD, it is possible to evaluate partial derivatives of numerical functions at given points. AD can be applied in two different modes, the forward and the reverse mode, where the forward mode has been shown to be well-suited and more easily applicable for large programs and a limited number of optimization parameters [19]. Hence, the forward version was applied in this work and will be introduced briefly. The interested reader is referred to the concise introduction by Griewank and Walther [32].

Together with the value of any variable $x \in \mathbb{R}$, the values of its partial derivatives with regard to parameters $\alpha_1, \ldots, \alpha_n$ are stored. Therefore, instead of a real number, x is considered as a tuple $\left[x, \frac{\partial x}{\partial \alpha_1}, \ldots, \frac{\partial x}{\partial \alpha_n} \right] \in \mathbb{R}^{n+1}$. Hence, the parameters α_i themselves can be written as $(\alpha_i, 0, \ldots, 0, 1, 0, \ldots, 0)$ with derivative $\frac{\partial \alpha_i}{\partial \alpha_i} = 1$ at the *i*-th component. Any arithmetic operation of two variables is then applied to the values and to their partial derivatives, according to the basic derivation rules, e.g., the result of a multiplication $x \cdot y$ is then

$$\left[x, \frac{\partial x}{\partial \alpha_1}, \ldots, \frac{\partial x}{\partial \alpha_n} \right] \cdot \left[y, \frac{\partial y}{\partial \alpha_1}, \ldots, \frac{\partial y}{\partial \alpha_n} \right] = \left[xy, x\frac{\partial y}{\partial \alpha_1} + y\frac{\partial x}{\partial \alpha_1}, \ldots, x\frac{\partial y}{\partial \alpha_n} + y\frac{\partial x}{\partial \alpha_n} \right].$$

In the framework of the open source CFD library OpenLB [33,34], AD was implemented via operator overloading in C++: for every arithmetic operation, the corresponding operation for the data type `ADf<double,n>` that stores value as well as partial derivatives was implemented. All functions and classes were templatized with regard to the underlying primitive data type so that they can be instantiated using the data type `ADf<double,n>` and then can yield function values as well as their partial derivatives.

The correct implementation of the `ADf`-type and its methods was guaranteed via unit tests. For a benchmark test (validation of AD-computed sensitivities), see Section 3.2.

2.6. Numerical Optimization

The computation of the variance $\sigma^2(\varphi)$ relies on the average $E(\varphi)$, which can only be computed at the end of the time interval. In order to avoid the need for storing the spatial concentration data for the computation of $\sigma^2(\varphi)$, we replaced $E(\varphi)$ by the average integral over the *penultimate* time period. A comparison (see Table 1) shows that the difference between these versions is low and that the trends of the two versions are the same.

Table 1. Average concentration $E(\varphi)$ over the two last time periods of length t_p and relative difference between these values, for varying of t_p.

Period Length t_p in s	$E(\varphi)$ over $[T - 2t_p, T - t_p]$	$E(\varphi)$ over $[T - t_p, T]$	Relative Difference
0.5	0.2188	0.2277	0.0392
0.6	0.2161	0.2271	0.0484
0.7	0.2132	0.2264	0.0582
0.8	0.2099	0.2255	0.0692
0.9	0.2060	0.2243	0.0818

The spatial integrals in the definitions of $E(\varphi)$, $\sigma(\varphi)$ were computed according to a first order quadrature rule (midpoint scheme) by summing the respective arguments for each mesh point, weighted by the voxel size. Accurate derivatives of the time integrals with regard to the period length t_p are essential for the optimization with regard to t_p, i.e., we need to capture the exact interval length, despite its limits not coinciding in general with discrete time steps. With a composite trapezoidal rule with summation over the discrete time steps and constant extrapolation for estimating the integral arguments at the interval bounds, we obtained sufficiently accurate values, as well as consistent derivatives (computed with AD).

Optimization was then executed utilizing the common steepest descent and the L-BFGS algorithms, with Armijo and Armijo-Wolfe-Powell step conditions, respectively [35,36].

3. Results

3.1. Simulation

The simulations were performed within the open source LBM-library OpenLB [33,34], employing the numerical methods specified in Section 2.3. The process parameters correspond to phosphate or ink diluted in water. They are similar to [10] and are shown in Table 2, where the Reynolds number is computed on the basis of the characteristic velocity and the hydraulic diameter. For the outflow, a parabolic profile with maximum velocity 3.32×10^{-2} m/s is implemented. At the first inlet, the pressure is fixed to the characteristic pressure. At the second inlet, the pressure agrees with this value for a static, passive micromixer and fluctuates by $\tilde{p}(t)$ around the characteristic pressure in the active, instationary case.

Table 2. Process parameters.

Parameter	Value
Char. velocity U	3.32×10^{-2} m/s
Hydraulic channel diameter L	0.00133 m
Reynolds number Re	44.1
Mass diffusivity D	1.0×10^{-9} m^2/s
Fluid density ρ	1000 kg/m^3
Viscosity ν	1.0×10^{-6} m^2/s
Péclet number Pe	44,156

Since multiple simulation runs are essential for the numerical optimization, a careful choice and not too large of a resolution is desirable. While a perfectly stable simulation could be achieved with $\Delta x = 15.83$ μm, $\Delta t = 47.69$ μs, a precise capturing of the trend of the objective could be achieved with a significantly lower numerical effort; see Figure 3. Since the minimizers agree very well for different resolutions, the choice of $\Delta x = 36.94$ μm, $\Delta t = 111.3$ μs seems appropriate, which yields a lattice relaxation time of 0.7745.
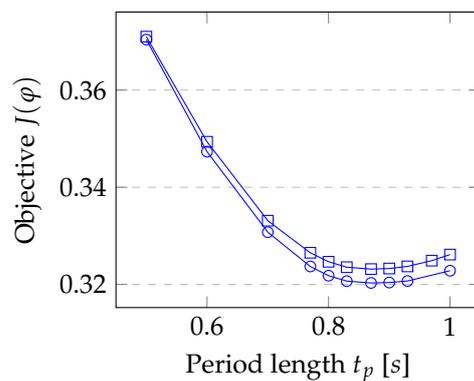
Figure 3. Selection of resolution. Segregation intensity for $\Delta x = 36.9$ μm (□) and $\Delta x = 27.7$ μm (○) and varying period length t_p, with $a = 1.2$, $d = 0.5$ and acoustic scaling ($\Delta t \sim \Delta x$). Although the values depend on the resolution, trend and location of the minimizer are identical for both curves.

Convergence with regard to the resolution Δx, Δt is checked for several quantities: the average velocity magnitude at the first inlet and $t = 4$ s, $y_1 := \|u(x, 4)\|_{L^2(\Gamma_I)}$, the L^2-norm of the concentration at the outlet at $t = 4$ s, $y_2 := \|\varphi(x, 4)\|_{L^2(\Gamma_{out})}$ and the average concentration and variance over the time interval [3.4 s, 4.0 s], $E(\varphi)$ and $\sigma^2(\varphi)$. These four quantities have been computed for different resolutions from $\Delta x = 36.94$ μm, $\Delta t = 111.3$ μs to $\Delta x = 12.31$ μm, $\Delta t = 37.09$ μs in acoustic scaling ($\Delta t \sim \Delta x$). The resulting relative errors with regard to the finest resolved simulation are shown in Figure 4. They show convergence of both fluid flow and concentration simulation.
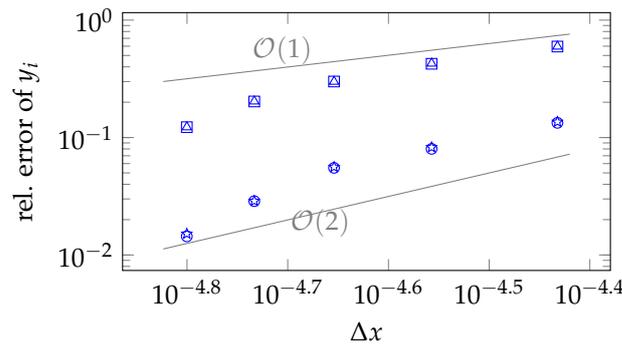
Figure 4. Grid independency study (simulations). Relative errors of $y_1 = \|u(x,4)\|_{L^2(\Gamma_l)}$ ($\square$), $y_2 = \|\varphi(x,4)\|_{L^2(\Gamma_{out})}$ ($\triangle$), $E(\varphi)$ ($\circ$), $\sigma^2(\varphi)$ ($\star$) with regard to the finest resolved simulation.

3.2. Simulation of Sensitivities

Next to the simulations themselves, the computation (simulation) of sensitivities has to be validated. This is carried out for various quantities of interest: fluid flow simulation and adsorptive simulation at both a fixed time and integrated over time.

First, the convergence of AD-computed sensitivities with regard to the resolution is checked for the quantities y_1, y_2, $E(\varphi)$, $\sigma^2(\varphi)$ as in the previous section, where the temporal integration domain for $E(\varphi)$ and $\sigma^2(\varphi)$ has been defined to [4.0 s $-t_p$, 4.0 s]. The relative errors of the t_p-sensitivities with regard to the finest resolved simulation are shown in Figure 5. Again, convergence becomes visible, where the convergence order of the sensitivities is similar to the convergence order of the corresponding values (see Figure 4).
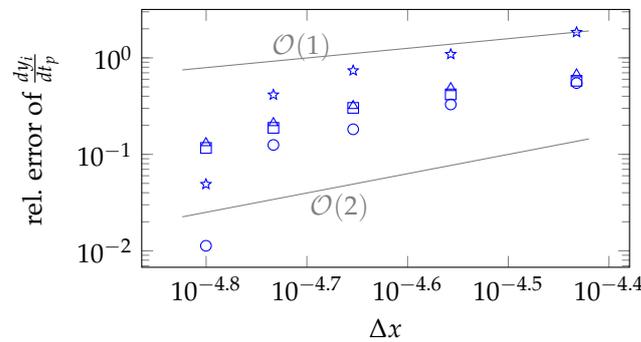
Figure 5. Grid independency study (sensitivities). Relative errors of sensitivity of $y_1 = \|u(x,4)\|_{L^2(\Gamma_l)}$ ($\square$), $y_2 = \|\varphi(x,4)\|_{L^2(\Gamma_{out})}$ ($\triangle$), $E(\varphi)$ ($\circ$), $\sigma^2(\varphi)$ ($\star$) with regard to t_p at $t_p = 0.6$ and the finest resolved simulation.

As a benchmark experiment, AD-computed sensitivities of different quantities of interest with regard to the control variables t_p, a and d are computed and compared to FDQ and CDQ of different step size h.

At the fixed time $t = 4$ s, we consider the L^2-norms of velocity u and concentration φ over Γ_l and Γ_{out}, respectively. As expected, the difference quotients converge to the AD-computed sensitivities as the step size h decreases (see Figure 6). The error stagnates at $h \approx 10^{-8}$ for FDQ (half of the machine precision) and $h \approx 10^{-6}$ for CDQ (one third of the machine precision), which can be explained by the well-known loss of accuracy that is observed for the numerical evaluation of difference quotients [31].
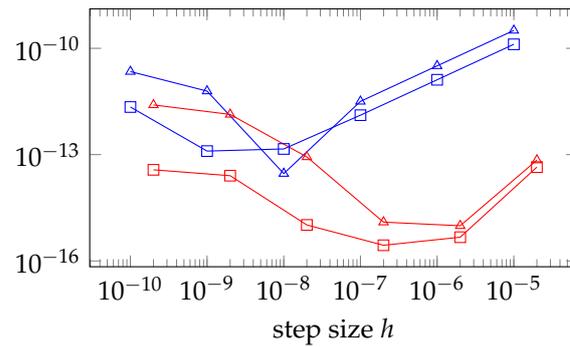
Figure 6. Accuracy of AD and DQ. Numerical error between difference quotients and AD-sensitivities of $y_1 = \|\boldsymbol{u}(\boldsymbol{x}, t = 4)\|_{L^2(\Gamma_l)}$ (FDQ □/CDQ □) and $y_2 = \|\varphi(\boldsymbol{x}, t = 4)\|_{L^2(\Gamma_{out})}$ (FDQ △/CDQ △) with regard to the period length $t_p = 0.6$ s, evaluated at $t = 4.0$ s. Blue color corresponds to FDQ, red to CDQ.

Furthermore, the integrals of φ and $\sigma^2(\varphi)$ over the outlet Γ_{out} and the time interval $[4 \text{ s} - t_p, 4 \text{ s}]$ are considered. We observe convergence of the difference quotients as the step size decreases (see Figure 7). The maximal accuracy has already been reached at around $h \approx 10^{-6}$ ($h \approx 10^{-5}$ for CDQ), which is probably due to a further loss of accuracy due to the time integration (the values for the different time steps have been added sequentially and a similar loss could be observed for the integration of analytical functions).
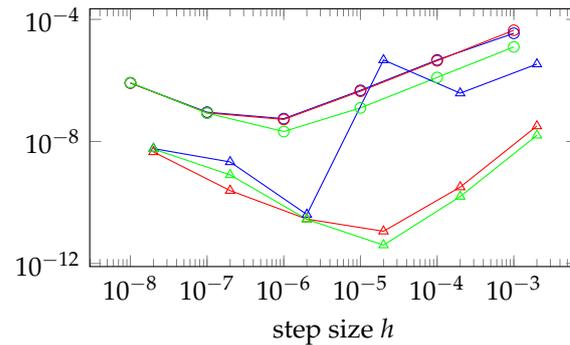
Figure 7. Accuracy of AD and DQ. Gap between difference quotients and AD-sensitivities of average concentration computed over the time interval $[4 \text{ s} - t_p, 4 \text{ s}]$ with regard to period length $t_p = 0.6$ s (FDQ ○/CDQ △, blue color), phase difference $d = 0.5$ (FDQ ○/CDQ △, red color) and amplitude $a = 0.5$ (FDQ ○/CDQ △, green color). Circles correspond to FDQ, triangles to CDQ. For the CDQ-sensitivity with regard to t_p, the number of time steps in the temporal integration domain changes if $h > 10^{-5}$, which introduces another source of numerical error and gives explanation for the outlier.

For all of these examples, the trends of the computed errors are those that are typically observed for the errors between difference quotients and exact or AD derivatives, so we consider the AD sensitivity computation as validated. The rather large residual error between AD and FDQ/CDQ, even for the optimal step sizes, can be interpreted as due to the limited applicability of FDQ/CDQ in this context and due to the accuracy gain by the utilization of AD. Another methodical aspect is that, with AD, the evaluation of an optimal step width becomes unnecessary.

3.3. Optimization

In addition to the global trend of the objective with regard to the variation of the period length (see Figure 3), the overall trends with regard to the amplitude and phase difference are computed; see Figure 8. For all of these variables of interest, the dependence of the objective is either monotone or convex; hence, the global optimization problem seems well-posed.
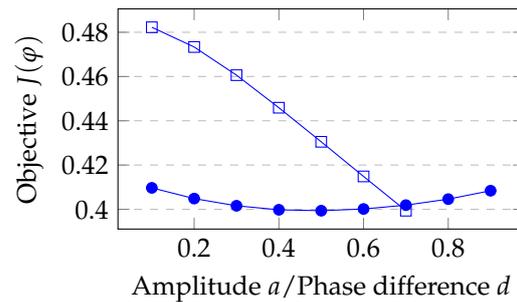
Figure 8. Influence of design parameters. Segregation intensity for $\Delta x = 36.9$ μm and varying amplitude/phase difference. Squares: $t_p = 0.8$ s, $d = 0.5$, a is varied. Circles: $t_p = 0.8$ s, $a = 0.7$, d is varied.

In particular, the monotone influence of the amplitude allows us to exclude this variable from numerical optimization, since its optimal choice is the maximal value, where no backflow happens at the inlets and a dependence of this value on the other free parameters does not seem relevant. Hence, numerical optimization has to be executed only with regard to the period length and phase difference, while the (almost optimal) choice $a = 0.5$ is fixed.

The optimization is then executed until the L^2-norm of the gradient $\left(\frac{dJ}{dt_p}, \frac{dJ}{dd} \right)$ is less than 6×10^{-6} and the relative change in the controlled parameters is less than 10^{-6}. For the composite cosine phase function p_{coscomp}, we find the optimum $J(\varphi) = 0.432266$ at $t_p = 0.7635904$ s, $d = 0.4986677$. In addition, the value of $J(\varphi)$ satisfied convergence in the sense that it changed by less that 10^{-10} in the last iteration step. The resulting optimum corresponds to an improved mixing quality of 14.3% over the static execution. Plots of velocity magnitude and adsorptive concentration are shown in Figures 9–11.

For the smoothed square wave phase function p_{ssqw}, the optimum is found at $J(\varphi) = 0.402634$ with $t_p = 0.7985104$ s, $d = 0.6528243$ and smoothing width $\epsilon = 0.01$. Again, the objective changes by less than 10^{-10} in the last optimization step. The improvement in mixing quality compared to static execution is 21.9%. For a smaller smoothing width $\epsilon = 10^{-3}$ in the definition of p_{ssqw}, the optimization becomes unstable. The probable cause is due to the gradient of the phase function not being resolved accurately then. On the one hand, the difference in the objectives is less than 0.01%. Thus, the size of ϵ is not really relevant for practical purposes. On the other hand, the slightly smaller objective for smaller ϵ, as well as the comparison of the results for p_{ssqw} and p_{coscomp}, fits to the expectation that higher concentration gradients increase the overall mixing performance.
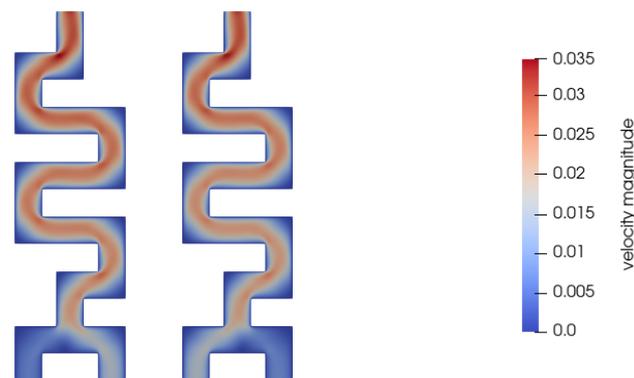
Figure 9. Flow velocity magnitude on a horizontal slice through the center of the channel. With composite cosine pulsation, at $t = 7.4$ s (**left**) and $t = 7.8$ s (**right**).
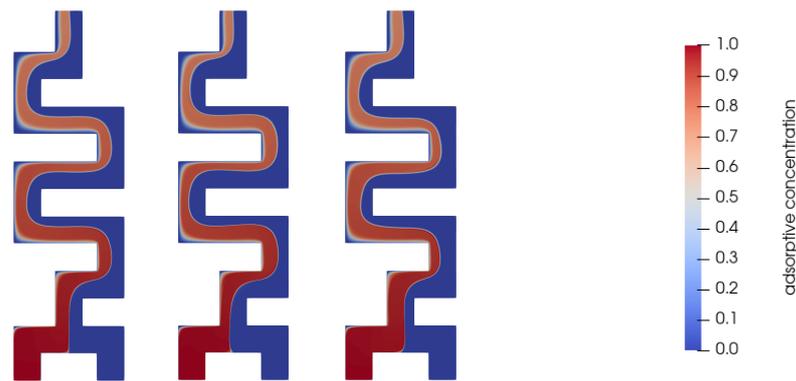
Figure 10. Adsorptive concentration on a horizontal slice through the center of the channel. Static regime (**left**), with composite cosine pulsation, at $t = 7.4$ s (**middle**) and $t = 7.8$ s (**right**).

Figure 11. Adsorptive concentration at the outlet. With composite cosine pulsation, at $t = 7.4$ s (**top**) and $t = 7.8$ s (**bottom**).

4. Discussion

It is noticeable that the established performance improvements via (optimal) pulsation differ quite largely among different studies (see [4,9,13]). We explain this by different underlying flow regimes, where the Reynolds and Péclet numbers vary by factors of ten and more. Indeed, in [9], a strong dependence of the (optimal) mixing quality on the Péclet and Reynolds number was found. Hence, a systematic investigation of this relation, including the numerical optimization of parameters, may be interesting to consider.

As an extension, the proposed methods may be applied to optimize features of the mixer geometry, which likely offers strong potential for a performance increase. Moreover, more advanced mixers, such as those using a fluid oscillator, may be considered.

5. Conclusions

In this work, a new approach for the numerical optimization of the pulsation of a micromixer has been presented and thoroughly validated. It is based on fluid flow and particle concentration simulations with LBM and the forward mode of AD for computing the derivatives. It is shown that both methods are well-suited for solving large-scale problems, i.e., their combination allows us to simulate partial derivatives of flow quantities with a very good accuracy. The approach is generic in the sense that flow simulation, AD and optimization algorithms are independent of each other, which allows us, e.g., to exchange the mixer geometry or to apply the same methods and framework to an entirely different application case. For the micromixer, the optimized pulsation results in a performance improvement of 21.9% with regard to the segregation intensity of the static run.

Author Contributions: Conceptualization, J.J. and M.J.K.; methodology, J.J.; software, J.J., J.E.M., L.H., J.M., F.B. and M.J.K.; validation, J.J., J.E.M. and L.H.; formal analysis, J.J.; investigation, J.J. and J.M.; resources, M.J.K.; data curation, J.J.; writing—original draft, J.J.; writing—review and editing, J.J., J.E.M., L.H. and M.J.K.; visualization, J.J.; supervision, J.J.; project administration, J.J.; funding acquisition, J.J. and M.J.K. All authors have read and agreed to the published version of the manuscript.

Nomenclature

Abbreviations:

AD	Automatic differentiation
ADE	Advection–diffusion equation
BGK	Bhatnagar–Gross–Krook
CDQ	Central difference quotient
CFD	Computational fluid dynamics
FDQ	Forward difference quotient
FEM	Finite element method
LBM	Lattice Boltzmann methods
NSE	Navier–Stokes equations

Constants, parameters and variables:

$\Delta t, \Delta x$	temporal and spatial resolution
$\Gamma_l, \Gamma_r, \Gamma_{out}, \Gamma_w$	boundary of the domain (left inlet, right inlet, outlet, wall)
Ω	flow channel
ϵ	smoothing width for $p_{\mathrm{ssqw}}(t)$
ν	kinematic viscosity
φ	relative mass concentration of adsorptive particles
ρ	fluid density
$\sigma^2(\varphi)$	variance of φ at the outlet over $[T - t_p, T]$
D	diffusion coefficient
$E(\varphi)$	average of φ at the outlet over $[T - t_p, T]$
$J(\varphi)$	segregation intensity at the outlet over $[T - t_p, T]$
$\tilde{J}(\varphi)$	mixing quality at the outlet over $[T - t_p, T]$
I	temporal simulation interval
L	hydraulic channel diameter
Pe	Péclet number
Re	Reynolds number
T	max. simulation time
U	char. fluid velocity
a	amplitude of inlet phase function
d	phase difference in inlet phase function
h	step width for FDQ/CDQ
p	kinematic pressure
$\tilde{p}(t)$	time-dependent pressure at the right inlet
$p_{\mathrm{coscomp}}(t)$	periodic inlet phase function: composite cosine
$p_{\mathrm{ssqw}}(t)$	periodic inlet phase function: smoothed square wave
$s(t)$	start-up function for velocity at the outlet
t	temporal coordinate
t_1, t_2	start-up time for outflow and pressure pulse, resp.
t_p	period length of inlet phase function
$\boldsymbol{u}, \boldsymbol{u_\tau}, u_n$	fluid velocity (3D vector, tangential to the boundary, normal to the boundary)
$\tilde{u}$	velocity profile at the outlet (normal component)
$\boldsymbol{x}$	spatial coordinates
y_1	L^2-norm of $\boldsymbol{u}$ at the left inlet at $t = 4$
y_2	L^2-norm of φ at the outlet at $t = 4$

References

1. Hardt, S.; Schönfeld, F. *Microfluidic Technologies for Miniaturized Analysis Systems*; Springer Science + Business Media, LLC: New York, NY, USA 2007; Volume 392. [CrossRef]
2. Hessel, V.; Löwe, H.; Schönfeld, F. Micromixers—A review on passive and active mixing principles. *Chem. Eng. Sci.* **2005**, *60*, 2479–2501. [CrossRef]
3. Raza, W.; Hossain, S.; Kim, K.Y. A Review of Passive Micromixers with a Comparative Analysis. *Micromachines* **2020**, *11*, 455. [CrossRef] [PubMed]
4. Glasgow, I.; Aubry, N. Enhancement of microfluidic mixing using time pulsing. *Lab Chip* **2003**, *3*, 114–120. [CrossRef] [PubMed]
5. Danckwerts, P.V. The Definition and Measurement of Some Characteristics of Mixtures. *Appl. Sci. Res. Sect. A* **1952**, *3*, 279–296. [CrossRef]
6. Khaydarov, V.; Borovinskaya, E.S.; Reschetilowski, W. Numerical and Experimental Investigations of a Micromixer with Chicane Mixing Geometry. *Appl. Sci.* **2018**, *8*, 2458. [CrossRef]
7. Ahmad Termizi, S.N.A.; Khor, C.Y.; Nawi, M.A.H.; Ahmad, N.; Ishak, M.; Rosli, M.; Jamalludin, M. Computational Fluid Dynamics (CFD) Simulation on Mixing in T-Shaped Micromixer. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *932*, 012006. [CrossRef]
8. Termizi, S.N.A.A.; Ishak, M.I.; Khor, C.Y.; Nawi, M.A.M.; Pouzay, M.F.B.M. Computational fluid dynamics (CFD) simulation on mixing in Y-shaped micromixer. *AIP Conf. Proc.* **2020**, *2291*, 020048. [CrossRef]
9. Rudyak, V.; Minakov, A. Modeling and Optimization of Y-Type Micromixers. *Micromachines* **2014**, *5*, 886–912. [CrossRef]
10. Maier, M.L.; Milles, S.; Schuhmann, S.; Guthausen, G.; Nirschl, H.; Krause, M.J. Fluid flow simulations verified by measurements to investigate adsorption processes in a static mixer. *Comput. Math. Appl.* **2018**, *76*, 2744–2757. [CrossRef]
11. Santana, H.S.; Silva, J.L.; Taranto, O.P. Optimization of micromixer with triangular baffles for chemical process in millidevices. *Sens. Actuators B Chem.* **2019**, *281*, 191–203. [CrossRef]
12. Hossain, S.; Ansari, M.; Husain, A.; Kim, K.Y. Analysis and optimization of a micromixer with a modified Tesla structure. *Chem. Eng. J.* **2010**, *158*, 305–314. [CrossRef]
13. Bockelmann, H.; Heuveline, V.; Barz, D.P.J. Optimization of an electrokinetic mixer for microfluidic applications. *Biomicrofluidics* **2012**, *6*, 024123. [CrossRef] [PubMed]
14. Bockelmann, H. High Performance Computing Based Methods for Simulation and Optimisation of Flow Problems. Ph.D. Thesis, Karlsruher Institut für Technologie, Karlsruhe, Germany, 2010. [CrossRef]
15. Gunzburger, M.D. *Perspectives in Flow Control and Optimization*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2002. [CrossRef]
16. Krause, M.J. Fluid Flow Simulation and Optimisation with Lattice Boltzmann Methods on High Performance Computers—Application to the Human Respiratory System. Ph.D. Thesis, Karlsruher Institut für Technologie, Karlsruhe, Germany, 2010.
17. Krause, M.J.; Heuveline, V. Parallel Fluid Flow Control and Optimisation with Lattice Boltzmann Methods and Automatic Differentiation. *Comput. Fluids* **2013**, *80*, 28–36. [CrossRef]
18. Nørgaard, S.; Sagebaum, M.; Gauger, N.; Lazarov, B. Applications of automatic differentiation in topology optimization. *Struct. Multidiscip. Optim.* **2017**, *56*. [CrossRef]
19. Zarth, A.; Klemens, F.; Thäter, G.; Krause, M.J. Towards shape optimisation of fluid flows using lattice Boltzmann methods and automatic differentiation. *Comput. Math. Appl.* **2021**, *90*, 46–54. [CrossRef]
20. Trunk, R.; Henn, T.; Dörfler, W.; Nirschl, H.; Krause, M.J. Inertial dilute particulate fluid flow simulations with an Euler–Euler lattice Boltzmann method. *J. Comput. Sci.* **2016**, *17*, 438–445. [CrossRef]
21. Krüger, T.; Kusumaatmaja, H.; Kuzmin, A.; Shardt, O.; Silva, G.; Viggen, E.M. *The Lattice Boltzmann Method—Principles and Practice*; Springer: Cham, Switzerland, 2016. [CrossRef]
22. Bhatnagar, P.L.; Gross, E.P.; Krook, M. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys. Rev.* **1954**, *94*, 511–525. [CrossRef]
23. Gruszczyński, G.; Dzikowski, M.; Wołłk, Ł.Ł. On recovering the second-order convergence of the lattice Boltzmann method with reaction-type source terms. *arXiv* **2021**, arXiv:2107.03962.
24. Schornbaum, F.; Rüde, U. Extreme-Scale Block-Structured Adaptive Mesh Refinement. *SIAM J. Sci. Comput.* **2018**, *40*, C358–C387. [CrossRef]
25. Courant, R.; Isaacson, E.; Rees, M. On the solution of nonlinear hyperbolic differential equations by finite differences. *Commun. Pure Appl. Math.* **1952**, *5*, 243–255. [CrossRef]
26. Latt, J.; Chopard, B.; Malaspinas, O.; Deville, M.; Michler, A. Straight velocity boundaries in the lattice Boltzmann method. *Phys. Rev. E* **2008**, *77*, 056703. [CrossRef] [PubMed]
27. Latt, J. Hydrodynamic Limit of Lattice Boltzmann Equations. Ph.D. Thesis, Université de Genève, Genève, Switzerland, 2007.
28. Skordos, P.A. Initial and boundary conditions for the lattice Boltzmann method. *Phys. Rev. E* **1993**, *48*, 4823–4842. [CrossRef] [PubMed]
29. Mangold, J. Optimierung Eines Statischen Mischers Mittels Kontrolle der Eingangsströmung. Bachelor's Thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2019.
30. Bothe, D.; Stemich, C.; Warnecke, H.J. Fluid mixing in a T-shaped micro-mixer. *Chem. Eng. Sci.* **2006**, *61*, 2950–2958. [CrossRef]
31. Sauer, T. *Numerical Analysis*, 2nd ed.; Featured Titles for Numerical Analysis; Pearson: London, UK, 2011.

32. Griewank, A.; Walther, A. *Evaluating Derivatives*, 2nd ed.; Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, USA, 2008. [CrossRef]

33. Krause, M.J.; Avis, S.; Kusumaatmaja, H.; Dapelo, D.; Gaedtke, M.; Hafen, N.; Haußmann, M.; Jeppener-Haltenhoff, J.; Kronberg, L.; Kummerländer, A.; et al. OpenLB Release 1.4: Open Source Lattice Boltzmann Code; Zenodo: Genève, Switzerland, 2020. [CrossRef]

34. Krause, M.J.; Kummerländer, A.; Avis, S.J.; Kusumaatmaja, H.; Dapelo, D.; Klemens, F.; Gaedtke, M.; Hafen, N.; Mink, A.; Trunk, R.; et al. OpenLB—Open source lattice Boltzmann code. *Comput. Math. Appl.* **2021**, *81*, 258–288. [CrossRef]

35. Geiger, C.; Kanzow, C. *Numerische Verfahren zur Lösung Unrestringierter Optimierungsaufgaben*; Springer: Berlin, Germany, 1999.

36. Byrd, R.H.; Lu, P.; Nocedal, J.; Zhu, C. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM J. Sci. Comput.* **1995**, *16*, 1190–1208. [CrossRef]