A Scalable Multiphysics Algorithm for Massively Parallel Direct Numerical Simulations of Electrophoresis

Dominik Bartuschat[a,*], Ulrich Rüde[a,b]

[a]*Lehrstuhl für Systemsimulation, Friedrich-Alexander Universität Erlangen-Nürnberg, Cauerstrasse 11, 91058 Erlangen, Germany*
[b]*Parallel Algorithms Group, CERFACS, 42 Avenue Gaspard Coriolis, 31057 Toulouse, France*

Abstract

In this article we introduce a novel coupled algorithm for massively parallel direct numerical simulations of electrophoresis in microfluidic flows. This multiphysics algorithm employs an Eulerian description of fluid and ions, combined with a Lagrangian representation of moving charged particles. The fixed grid facilitates efficient solvers and the employed lattice Boltzmann method can efficiently handle complex geometries. Validation experiments with more than $70\,000$ time steps are presented, together with scaling experiments with over $4 \cdot 10^6$ particles and $1.96 \cdot 10^{11}$ grid cells for both hydrodynamics and electric potential. We achieve excellent performance and scaling on up to $65\,536$ cores of a current supercomputer.

Keywords: Parallel simulation; electrokinetic flow; electrophoresis; fluid-particle interaction; MPI.

1. Introduction

1.1. Motivation

The motion of charged particles in fluids under the influence of electric fields occurs in a wide range of industrial, medical, and biological processes. When the charged particles are immersed in liquids, their migration caused by electric fields is termed electrophoresis. Due to the complex interplay of the physical effects involved in such particle-laden electrokinetic flows, numerical simulations are required to analyze, predict, and optimize the behavior of these processes. To this end, we present a parallel multiphysics algorithm for direct numerical simulations of electrophoretic particle motion.

Industrial applications that involve electrophoretic effects are electrofiltration [1, 2, 3] and electro-dewatering [4]. Moreover, electrophoresis is utilized in electrophoretic deposition techniques for fabricating advanced materials [5] and especially ceramic coatings [6, 7] in material science. Electrophoresis and electric fields are also applied in many medical and biological applications. The trend towards miniaturization of analysis processes has lead to the development of micro total analysis systems. Due to their high portability, reduced costs, fast operation, and high sensitivity [8, 9], the design of such lab-on-a-chip systems has been a highly active area of research for many years. These microfluidic systems require only small samples of liquid and particles, which are transported, manipulated, and analyzed in structures of length scales from several

---

*Corresponding author
Email address: dominik.bartuschat@cs.fau.de (Dominik Bartuschat)

nm to 100 µm. Therefore, microfluidic separation and sorting of particles and cells are important steps of diagnostics in such systems [10, 11]. Many of the employed techniques utilize electric fields to manipulate, separate, and sort biological particles and macromolecules [8], such as cells [9, 10] or DNA [12].

At the small scales of microfluidic analysis systems, flow measurements are difficult or even impossible. Moreover, the complex coupling of hydrodynamic and electrostatic effects involved in electrophoretic processes make predictions of electrophoretic motion challenging, especially for large numbers of particles. Therefore, numerical simulations are essential to aid the design and optimization of electrophoretic systems. The different physical effects in electrophoretic deposition can be better understood from insight gained in simulations. By means of such simulations, electrophoretic sorting in lab-on-a-chip systems can be optimized for maximal throughput, sorting efficiency, and sorting resolution. A review of simulation methods for electrophoretic separation of macromolecules is given in [13]. Also industrial applications of electrophoretic deposition can be optimized with the help of simulations, as presented in [14] for a coating process.

1.2. Multiphysics Coupling Strategy

For simulations of electrokinetic flows with electrophoretic particle motion, the coupling between three system components must be modeled: charged objects, fluid flow, and electric effects. The interacting physical effects are sketched in Fig. 1. In the simulation method introduced in this article, the motion of the
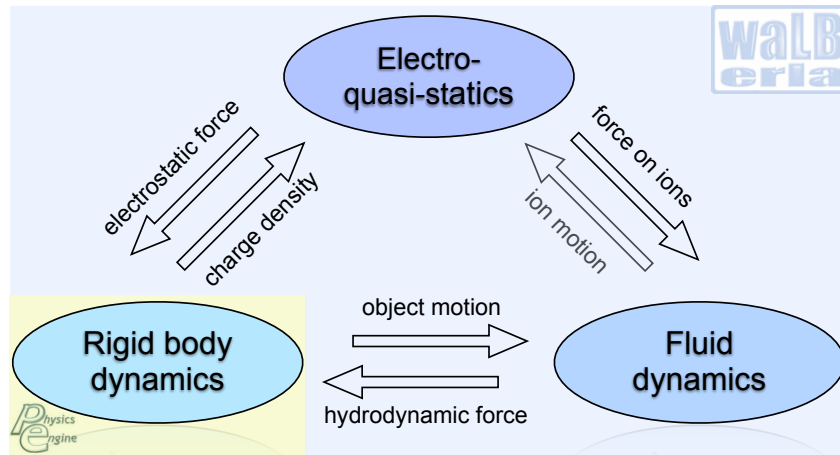
Figure 1: Coupled physical effects of electrophoresis simulated with WALBERLA and the physics engine *pe*.

rigid, charged particles is modeled with Newtonian mechanics. The motion of the surrounding fluid, which exerts hydrodynamic forces on the particles, is described by the incompressible Navier-Stokes equation. To capture fluid-particle interactions, the hydrodynamic forces and the influence of the particle motion on the fluid are modeled, based on the momentum exchange between fluid and particles. In this way, long-range hydrodynamic interactions between individual particles and between particles and walls are recovered.

Moreover, electrostatic forces exerted by applied electric fields on the charged particles are modeled, which are the main cause of electrophoretic motion. The varying positions of the charged particles in return affect the electric potential distribution in the simulation domain, based on their surface charge. Such a charge is carried by most biomolecules such as cells, proteins, and DNA [15]. In fact, most substances acquire surface charges when they get into contact with an aqueous medium [16] or electrolyte solution [17].

Electrostatic forces additionally act on the ions in the fluid and affect the fluid motion via body forces present in locations of net charge. This net charge originates from the repulsion of co-ions in the fluid of the same polarity as the surface charge and from the attraction of counter-ions. The ions in the fluid are transported with the fluid flow, which in turn alters the electric potential distribution. Ion motion in the fluid due to an electric field is governed by the Nernst-Planck equation, an advection-diffusion equation that employs a continuum description and treats the ions as point charges. For flows in which diffusion strongly dominates

over advection and therefore quasi-thermodynamic equilibrium can be assumed to hold, as considered in this paper, the electric potential due to the ion charge distribution is governed by the Poisson-Boltzmann equation.

The region of the particle's surface charge and of excess counter-ions in the fluid is denoted as *electric double layer* (EDL). According to the Stern model [18] employed in this article, this double layer comprises a region of ions attached to the surface and a diffuse part in which the ion concentration follows a Boltzmann distribution. At the surface of shear between the particle and the surrounding diffuse double layer, the characteristic ζ-potential is defined. The employed equilibrium considerations based on the Poisson-Boltzmann equation capture the dominant retardation effect of electrophoretic motion. This effect describes the retardation of the charged particle motion by the action of the applied electric field on the opposite net charge in the surrounding EDL. At high ζ-potentials, additionally the weaker relaxation effect occurs [16] that is caused by a distortion of the EDL and can be captured by the Nernst-Planck equation.

In this article, we present an efficient parallel multiphysics simulation algorithm for electrophoresis on a fixed Eulerian grid with a Lagrangian representation of moving particles. The particles are represented by the physics engine *pe* [19, 20] as geometrically fully resolved three-dimensional objects. Dependent on the electrostatic and hydrodynamic forces acting on the particles, the *pe* computes their trajectories by rigid body dynamics and additionally resolves particle collisions. The *pe* is coupled to waLBerla [21, 22, 23], a massively parallel simulation framework for fluid flow applications that employ the lattice Boltzmann method (LBM) [24, 25]. By means of a modular software structure that avoids dependencies between modules, functionality from different modules can be combined flexibly. To model fluid-particle interactions, the LBM momentum exchange method [26, 27] implemented in waLBerla [28] is applied. For the electrophoresis simulations, the LBM is performed with the two-relaxation time collision operator [29, 30] and an appropriate forcing term for the electric body force due to the ions in the fluid. The electric potential is represented by the Debye-Hückel approximation of the Poisson-Boltzmann equation that is discretized with finite volumes whose mesh structure naturally conforms to the lattice Boltzmann (LB) grid. This discretization also facilitates accommodating variable and discontinuous dielectricity values that vary in time according to the particle positions, as required for simulating dielectrophoretic effects. By means of the waLBerla solver module introduced in [31] together with the cell-centered multigrid solver implemented therein, the resulting linear system of equations is solved. Since the counter-ions lead to a quicker decay of the electric potential compared to the long-range electric potentials modeled in [31], the parallel successive over-relaxation (SOR) method implemented in this module is an adequate choice.

In a previous article, we have shown that the implemented fluid-particle interaction algorithm for arbitrarily shaped objects efficiently recovers hydrodynamic interactions also for elongated particles [32]. Moreover, we presented a parallel multiphysics algorithm for charged particles in the absence of ions in the fluid in [31]. We have therein shown that several millions of charged particles with long-range hydrodynamic and electrostatic interactions can be simulated with excellent parallel performance on a supercomputer. The present paper extends these simulation algorithms by also considering ions in the EDL around the particles and their effect on the fluid motion, and presenting suitable parallel coupling techniques. Together with the full four-way coupling of the fluid-particle interaction [31], the coupling with the quasi-equilibrium representation of the electric effects results in a 7.5-way interaction.

1.3. Related Work

In the following, we give an overview of numerical methods for simulations of electrophoretic phenomena that have been developed for different resolution levels. At the coarsest modeling scale, both fluid and solid phase are described by an Eulerian approach. These continuum models represent the charged species in terms of concentrations and are well suited for simulations with large numbers of unresolved particles. In [33] two-dimensional electrophoresis simulations of biomolecule concentrations are presented. These finite element simulations consider the effect of reactive surfaces and the electric double layer is included through the biomolecule charge. Also three-dimensional parallel simulations of electrophoretic separation with continuum approaches have been reported. The finite element simulations in [34] consider the buffer composition and the ζ-potential at channel walls. In [35, 36] mixed finite element and finite difference simulations of protein separation were performed on up to 32 processes.

At the finest level of resolution, fluid, ions, and particles are simulated by Lagrangian approaches. These explicit solvent methods [37] typically apply coarse-grained molecular dynamics (MD) models to describe the motion of fluid molecules and incorporate Brownian motion [38]. The mesoscale *dissipative particle dynamics* method is applied in [39] to simulate electrophoresis of a polyelectrolyte in a nanochannel. Another explicit solvent method is presented in [40] for simulating DNA electrophoresis, modeling DNA as a polymer. In both methods the polymer is represented by bead-spring chains with beads represented by a truncated Lennard-Jones potential and connected by elastic spring potentials. Explicit solvent models, however, are computationally very expensive, especially for large numbers of fluid molecules due to pairwise interactions [40]. Moreover, the resolution of solvent, macromolecules, and ions on the same scale limits the maximal problem sizes that can be simulated [41]. Also the mapping of measurable properties from colloidal suspensions to these particle-based methods is problematic [37].

The high computational effort is significantly reduced in implicit particle-based methods that incorporate hydrodynamic interactions into the inter-particle forces. Such methods are applied in [37] and [42] to simulate electrophoretic deposition under consideration of Brownian motion and van der Waals forces. Nevertheless, these methods are restricted to few particle shapes and hydrodynamic interactions in Stokes flow.

Euler-Lagrange methods constitute the intermediate level of resolution. These approaches employ Eulerian methods to simulate the fluid phase, whereas the motion of individual particles is described by Newtonian mechanics. For simulations of particles in steady-state motion, the resolved particles can be modeled as fixed while the moving fluid is modeled by an Eulerian approach. In [43] the finite volume method is applied to simulate electrophoresis of up to two stagnant toroids in a moving fluid, employing the Hückel approximation for a fixed ζ-potential and different electrical double layer thicknesses. The steady-state electrophoretic motion of particles with low surface potentials under a weak applied electric field in a charged cylindrical pore is simulated in [44] for a single cylinder and in [45] for two identical spheres. In both cases, a two-dimensional simulation with a finite element method for Stokes flow and Hückel approximation is performed, exploiting the axial symmetry of the problem.

For electrophoresis at steady state *perturbation approaches* can be employed that are based on the assumption that the double layer is only slightly distorted from the equilibrium distribution for weak applied electric fields (w.r.t. the field in the EDL). In addition to the equilibrium description based on the Poisson-Boltzmann equation, small perturbations in the equilibrium EDL are considered in terms of linear correction terms in the applied electric field for the ion distribution and the electric potential (see e.g. [46]). Using a perturbation approach with finite elements, the electrophoresis of two identical spheres along the symmetry axis of a cylindrical domain at pseudo steady-state were studied in [47]. Additionally to the hydrodynamic and electric interactions, these axisymmetric simulations consider van der Waals forces for particles in close proximity. In [48] a perturbation approach is applied to simulate a single colloid in a rest frame with periodic boundary conditions (BCs). The zeroth-order perturbation corresponds to the Poisson-Boltzmann equation, which is solved by a constrained variational approach suggested in [49]. For the first-order perturbation, the stationary Stokes equation is solved by a surface element method, the convection-diffusion for ionic concentrations by a finite volume solver, and the Poisson equation by a fast Fourier Transform method.

More sophisticated Euler-Lagrange methods include direct numerical simulation (DNS) models that represent the moving particles as geometrically fully resolved objects. These methods for particulate flows include approaches with *body-fitted moving meshes* and *fixed meshes*. Moving meshes can be represented by the Arbitrary-Lagrangian-Eulerian (ALE) formulation [50, 51] that employs moving unstructured meshes for fluid-particle interaction problems. Such an ALE method is applied in [52] to simulate electrophoresis of a single particle surrounded by a thin electrical double layer. The moving-mesh techniques require re-meshing when the distortion of the updated mesh becomes too high, and subsequent projection of the solution onto the new mesh. This overhead is circumvented in *fixed-mesh techniques* that allow the use of regular grids and therefore the application of efficient solvers. The *fluid particle dynamics* method [53] falls into the latter category, solving the Navier-Stokes and continuity equation on a fixed lattice, and representing the moving solid particles by fluid particles of very high viscosity. By means of a concentration field that represents the particle distribution, the particles affect the fluid viscosity and, together with forces acting on the particles, the body force term of the Navier-Stokes equation. The rigid particles are geometrically modeled by the Lennard-Jones potential and their motion is described by Newtonian mechanics [53]. This technique is

applied to simulate electrophoretic deposition of two charged particles in [54] and electrophoretic separation in [55]. In these simulations the electrostatic interactions of the particles are modeled in terms of the body force field, together with the advection and diffusion of ions and the resulting effect of the applied electric field on the fluid motion. With this method, however, the particle rigidity is imposed by very high viscosity values that restrict the time step size [56] and the Lennard-Jones potential restricts the particles shapes to spheres. The *smoothed profile method* (SPM) [56] circumvents this time-step constraint by directly modeling the particles as solid objects. Inside the particles and at solid-fluid boundaries that are represented by diffuse interfaces, a body force is imposed on the fluid to model the effect of the particle motion on the fluid. The fluid is again modeled on a fixed Cartesian grid and the particle motion with Newtonian mechanics, where particle overlaps are typically prevented by a truncated Lennard-Jones potential. With this method, electrophoresis of charged spherical particles is simulated in [57] for a constant, uniform electric field and in [58] for an oscillating electric field. In both articles, the ion number concentration is modeled by an advection-diffusion equation to recover non-equilibrium double layer effects. The SPM is also applied in [59] to simulate electrophoresis of single cylinders and microtubules, employing the equilibrium representation of the EDL. A further fixed-mesh technique is the *immersed boundary method* [60, 61, 62], where the rigid body motion is imposed on the flow by body forces applied at the particle boundaries. This method, combined with a finite volume method for solving the steady-state Poisson-Nernst-Planck equation system, is applied in [63] to simulate the electrophoretic motion of up to three spherical particles in a two-dimensional setup.

Lattice-Boltzmann based methods are very well suited for parallel direct numerical simulations of fluid-particle interactions on fixed Cartesian grids. Both the Lagrangian particles and ions are often explicitly modeled by molecular dynamics approaches that represent the rigid objects by repulsive potentials. In [64] the electrophoresis of a colloidal sphere immersed in a fluid with counter-ions is simulated, modeling the solvent by a lattice Boltzmann method. The charged sphere modeled with molecular dynamics is represented by a raspberry model that comprised several beads connected by the finitely extensible nonlinear-elastic (FENE) potential. Using a modified raspberry model with two spherical shells of beads solidly attached to a larger spherical particle, this method is extended in [65] to simulate the electrophoresis of a spherical Janus particle. The partially uncharged particle is surrounded by anions and cations represented by charged beads. Electrophoresis simulations for a single highly charged spherical macro-ion in an electrolyte solution with explicitly modeled positive and negative micro-ions are presented in [66]. Since the coupling of fluid and macro-ion is performed via several particle boundary points, a single spherical particle is sufficient to represent the macro-ion. A similar LB-MD method is applied in [67] to simulate the stretching of a charged polyelectrolyte between parallel plates. The polyelectrolyte immersed in a liquid with explicitly modeled counter- and co-ions is modeled by beads bonded together by the FENE potential. In all these LB-MD simulations with explicitly modeled ions, hydrodynamic interactions are simulated with the LBM and thermal fluctuations are added to both the fluid and the MD objects. The high computational effort for modeling each individual ion by means of molecular dynamics, however, restricts the maximum feasible problem size. With these approaches, only a limited number of ions per colloidal particle can be simulated and the colloid radius is typically restricted to one order of magnitude larger than the ion size [48]. Therefore, approaches based on continuum descriptions of the suspended ions are more practical for simulations of many or for larger charged particles.

Alternatively to the continuum approach based on the Poisson-Boltzmann equation employed in this article, electrophoresis can be simulated with the *link-flux method* [68] that models the advection and diffusion of ions in terms of Nernst-Planck equations. The link-flux method employs the LBM for fluid dynamics and models ion motion in terms of fluxes between lattice cells. In [69] this method is compared to a LB-Poisson-Boltzmann approach for a fixed spherical particle in a periodic three-dimensional domain. The aim is to examine the influence of particle motion and counter-ion concentration on the ζ-potential, leading to the conclusion that for weakly perturbing electric fields or low Péclet numbers the equilibrium and dynamic ζ-potentials are indistinguishable. The link-flux method is extended in [41] to support moving particles in combination with the LB momentum exchange method. To ensure charge conservation in electrophoresis simulations, appropriate moving BCs for the solute fluxes are introduced. This method is verified in [41] by electrophoresis simulations of up to eight particles.

1.4. Objectives and Outline

The primary goal of this paper is the introduction of a parallel multiphysics algorithm for electrophoresis simulations together with validations of the physical correctness of the coupled algorithm for different particle sizes. For this algorithm, WALBERLA is augmented by an efficient boundary handling method that is able to treat electric potential BCs on the moving particles. Moreover, a joint parameterization for the different coupled numerical methods is introduced. To achieve excellent computational performance, a matrix-free representation of the linear system based on a *stencil* paradigm is used in the solver module [31]. For the linear Debye-Hückel approximation it is systematically exploited that these stencils are almost uniformly identical throughout the simulation domain. The validation runs were performed on up to 8192 parallel processes of a modern supercomputer. Moreover, simulation results for the electrophoretic motion of a single particle in a micro-channel are presented, including visualizations of the electric potential distribution and of the resulting flow field around the particle. Finally, we present performance results on a modern supercomputer. Very good scaling behavior is shown for more than $196 \cdot 10^9$ lattice cells on up to $65\,536$ cores. In this case, more than four million fully resolved charged objects and their surrounding double layers are simulated, together with the interactions with the fluid. To the best of our knowledge, electrophoresis simulations of that size are unprecedented.

The equilibrium considerations in the present paper recover the predominant retardation effect due to an opposing electrostatic force on the net opposite charge in the electrical double layer that counteracts the particle motion. For the presented method, a computationally cheap and flexible SOR method is sufficient to solve the electric potential equations. With our approach we aim for simulations of millions of charged particles as in [31]. For these large numbers of particles the dynamics of an electrical double layer as in [41] is computationally too expensive, even on modern supercomputers.

This paper is structured as follows: The physical background of fluid-particle interactions and electrophoresis are described in Sec. 2 and Sec. 3, respectively. In Sec. 4, the employed LB-momentum exchange method for fluid-particle interactions is outlined, together with the finite volume discretization and the common parameterization concept for the coupled multiphysics methods. Then the extension of the WALBERLA framework for the electrophoresis algorithm is described in Sec. 5. Finally validation results for the electrophoretic motion of a spherical particle and visualizations of the resulting flow field and electric potential distribution are presented in Sec. 6 before conclusions are drawn in Sec. 7.

2. Fluid-Particle Interaction

The macroscopic description of fluid behavior is based on the continuum hypothesis (cf. Batchelor [70]) that allows to consider a fluid as continuum. Fluid properties are then represented by the macroscopic quantities density ρ_f, velocity $\vec{u}$, and pressure p, as functions of space and time. In terms of these quantities, fluid dynamics is described by conservation laws for mass, momentum, and energy. In this article isothermal flows are considered, and therefore the energy equation does not have to be solved. Moreover, non-continuum effects that become relevant for gas flows at very small scales [71] are assumed to be negligible. Therefore, no-slip BCs are applied at solid-fluid interfaces, and slip velocities due to non-continuum Knudsen layer effects are not considered.

Conservation of mass is described by the continuity equation. For incompressible fluids the density is not affected by pressure changes [70, 72] and the continuity equation reads

$$\nabla \cdot \vec{u} = 0. \tag{1}$$

Conservation of momentum in a viscous, compressible fluid can be described in terms of the momentum flux density tensor $\underline{\underline{\Pi}}$ [72]. The temporal change of momentum in an Eulerian control volume is balanced by the net momentum flux through its surface and by external body forces $\vec{f_b}$ acting on the volume as

$$\frac{\partial \left(\rho_f \vec{u} \right)}{\partial t} = -\nabla \cdot \underline{\underline{\Pi}} + \vec{f_b}. \tag{2}$$

The second-order tensor $\underline{\underline{\Pi}}$ comprises a term for the convective transport of momentum and the total stress tensor $\underline{\underline{\sigma}}$ for the momentum transfer due to pressure and viscosity

$$\underline{\underline{\Pi}} = \rho_f \, \vec{u}\vec{u}^\top - \underline{\underline{\sigma}}. \tag{3}$$

The total stress tensor $\underline{\underline{\sigma}}$ can be decomposed into a part representing normal stresses related to the pressure and a viscous part related to shear stresses. For incompressible Newtonian fluids, the stress tensor reads as

$$\underline{\underline{\sigma}} = -p\underline{\underline{I}} + \mu_f \left[\nabla\vec{u} + (\nabla\vec{u})^\top \right], \tag{4}$$

where the first term with the second-rank identity tensor $\underline{\underline{I}}$ contains the thermodynamic pressure p defined according to Landau & Lifshitz [72] as used in the LBM literature [73]. The second term with dynamic viscosity μ_f represents the shear stresses that are proportional to the rate of deformation [74]. With this stress tensor the incompressible Navier-Stokes equation results from Eqn. (2) with the continuity equation for compressible fluids [75] as

$$\underbrace{\rho_f \left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} \right)}_{\text{inertial forces}} = \underbrace{-\nabla p}_{\text{pressure stress}} + \underbrace{\mu_f \, \Delta \, \vec{u}}_{\text{viscous stress}} + \underbrace{\vec{f_b}.}_{\text{external body force}} \tag{5}$$

The left-hand side represents inertial forces, and the right-hand side describes surface forces equivalent to stress in the fluid [70] and body forces. These volume forces include gravity or electrostatic force and are represented as force per unit volume $\vec{f_b}$.

An important dimensionless quantity to characterize fluid flows is the Reynolds number $Re = \frac{UL}{\nu_f}$, with kinematic viscosity $\nu_f = \frac{\mu_f}{\rho_f}$ and characteristic velocity and length scale, U and L. In the creeping motion regime ($Re \ll 1$), inertial forces are negligible, and the Stokes equations for incompressible Newtonian fluids result as

$$\begin{aligned} -\nabla p + \mu_f \, \Delta \, \vec{u} + \vec{f_b} &= 0 \\ \nabla \cdot \vec{u} &= 0. \end{aligned} \tag{6}$$

For these linear equations the superposition principle holds, which is often utilized for fluid-particle interaction in Stokes flow and is employed in Sec. 6.4.

Particles immersed in a fluid experience a force in case of relative fluid motion or a pressure gradient in the fluid. This force can be calculated from the stresses in the fluid next to the particle by integrating the stress tensor $\underline{\underline{\sigma}}$ over the particle surface Γ_p [74, 46]

$$\vec{F}_{\text{part}} = \int_{\Gamma_p} \underline{\underline{\sigma}} \cdot \vec{n} \, dA. \tag{7}$$

Here, dA denotes the surface area elements and $\vec{n}$ the associated normal vector. For simple cases, such as spherical bodies moving in unbounded Stokes flow, or single bodies moving in confined domains, analytical solutions for the particle motion are known as described in the following. More complex fluid-structure interaction problems must be solved numerically, e. g. by the LBM.

For the computation of particle motion in incompressible fluids, hydrostatic effects that result e. g. from gravity, do not have to be considered explicitly in the momentum equation. In this case, the force exerted by the fluid on the particle according to Eqns. (7) and (4) is the *drag force* given by

$$\vec{F}_d = - \int_{\Gamma_p} p \, \vec{dA} + \mu_f \int_{\Gamma_p} \nabla\vec{u} + (\nabla\vec{u})^\top \, \vec{dA},$$

where p is the hydrodynamic part of the total pressure.

The resistance to the motion of a sphere in an unbounded fluid at very low Reynolds numbers can be calculated from the above expression for the drag force and the Stokes equations (6). For a sphere located at $\vec{x} = \vec{0}$, the unbounded fluid is represented by

$$\vec{u} \to 0 \quad \text{as} \quad \vec{x} \to \infty$$

imposed on the fluid velocity in the Stokes equations. The resulting drag force

$$\vec{F}_d = -6\pi\mu_f R\vec{U}. \tag{8}$$

acting on a rigid sphere of radius R and velocity $\vec{U}$ was derived by Stokes [76]. This equation is commonly referred to as *Stokes' law*.

For a sphere moving in a fluid subject to a constant force, Stokes' law relates the terminal steady-state velocity of the sphere to the drag force exerted by the fluid. Such a force may e.g. be the Coulomb force acting on a charged particle. The terminal sphere velocity is then obtained from the balance of the external force and the drag force, $\vec{F} + \vec{F}_d = \vec{0}$, as

$$\vec{U} = \frac{1}{6\pi\mu_f R}\vec{F}. \tag{9}$$

A particle moving in a confined domain experiences a retardation caused by surrounding walls. The effect of walls on a moving particle in Stokes regime can be determined by means of the method of reflections, as described in detail in [74]. Happel & Bart [77] employed this method to obtain a first-order correction to the drag force on a sphere settling in a long square duct with no-slip walls. Miyamura et al. [78] found polynomial expressions for the increased drag by fitting the coefficients to experimentally obtained settling velocities of spheres in different confining geometries. The correctness of the wall effect recovered in LBM simulations with the fluid-particle interaction algorithm employed in this article was verified in [75] against these expressions.

3. Electrokinetic Flows

3.1. Governing Equations for Electrokinetics

The transport of ions in fluids subject to electric fields that occurs in electrokinetic flows can be modeled by means of a continuum theory, similar to fluid dynamics. Instead of modeling individual ions and their interactions, local ion concentrations n_i and fluxes $\vec{j}_i$ of the different ionic species i are considered. Based on these macroscopic quantities the ion transport in dilute electrolyte can be described by the law for the conservation of ionic species in a solution

$$\frac{\partial n_i}{\partial t} = -\nabla \cdot \vec{j}_i \tag{10}$$

in the absence of chemical reactions. Here n_i denotes the number density or concentration that is related to the molar concentration as $c_i = n_i/N_A$ with Avogadro's number N_A. The total ionic flux $\vec{j}_i$ of species i comprises an advective flux with a common mass average velocity $\vec{u}$ for all species and fluxes relative to the advective flux due to diffusion and electric migration [46]. This relation is expressed by the *Nernst-Planck* equation

$$\vec{j}_i = \underbrace{n_i\vec{u}}_{\text{advective flux}} - \underbrace{D_i\nabla n_i}_{\text{diffusive flux}} - \underbrace{n_i\mu_i^*\nabla\Phi}_{\text{migration flux}}, \tag{11}$$

where D_i and μ_i^* represent the spatially homogeneous diffusion coefficient and ionic mobility of species i, respectively, and $\nabla\Phi$ the local electric potential gradient. The ionic mobility is defined as $\mu_i^* =: \frac{D_i z_i e}{k_B T}$, where e denotes the elementary charge, z_i the valence of a given ionic species, k_B the Boltzmann constant and T the temperature.

To model the influence of the charged ions on the electric potential governed by the Poisson equation

$$-\Delta\Phi(\vec{x}) = \frac{\rho_e}{\varepsilon_e} \tag{12}$$

8

for spatially uniform fluid permittivity ε_e, the ion charge distribution is considered in terms of the local mean macroscopic charge density as

$$\rho_e = \sum_i e\, z_i\, n_i. \tag{13}$$

The Poisson-Nernst-Planck system Eqns. (10)–(12) is highly nonlinear, and solving the overall system is computationally very expensive, especially for electrophoresis of many particles. Therefore the problem is simplified by restriction to equilibrium considerations based on the Boltzmann distribution that capture the dominant electrophoretic effects. The resulting Poisson-Boltzmann equation holds for (quasi-)thermodynamic equilibrium when the ion distribution is not affected by fluid flow or by externally applied electric fields.

For the fluid this means that flows must be diffusion-dominated, characterized by very small Péclet numbers $Pe = \frac{UL}{D}$ for mass transfer. These dimensionless numbers relate the advection rate of a fluid to its diffusion rate for a given flow speed U, length scale L (i.e., particle radius), and diffusion coefficient D. Additionally the applied electric field must be weak compared to the potential difference across the diffuse layer of characteristic thickness or Debye length λ_D, i.e., $E_{\text{ext}} \ll \frac{\zeta}{\lambda_D}$ [79].

Based on these assumptions the electric potential ψ resulting from the non-uniform ion distribution in the EDL is considered in the following, instead of the total electric potential $\Phi = \psi + \varphi$ that additionally comprises the potential φ of the externally applied electric field.

The Boltzmann distribution for ions can be derived from the Nernst-Planck equation, as outlined in [46]. Considering the Nernst-Planck equation (11) in one dimension at equilibrium and integrating from a reference point in the bulk with potential ψ_∞ and concentration $n_{i\infty}$, yields

$$n_i = n_{i\infty}\, e^{-\dfrac{z_i\, e(\psi - \psi_\infty)}{k_B\, T}}. \tag{14}$$

Setting the reference potential ψ_∞ in the electroneutral bulk solution to zero recovers the Boltzmann distribution with the number density $n_{i\infty}$ at the location of the neutral state.

From Poisson's equation (12) with net charge density as in Eqn. (13) and the obtained Boltzmann distribution, the Poisson-Boltzmann equation follows as

$$-\Delta\psi = \frac{e}{\varepsilon_e} \sum_i z_i n_{i\infty} e^{-\dfrac{z_i\, e\, \psi}{k_B\, T}}, \tag{15}$$

relating the electric potential ψ to the ion concentrations at equilibrium. For binary, symmetric electrolyte solutions comprising two species of valence $z = -z_- = z_+$, the Poisson-Boltzmann equation takes the form

$$-\Delta\psi = -\frac{2\, z\, e\, n_\infty}{\varepsilon_e} \sinh\left(\frac{z\, e\, \psi}{k_B\, T}\right). \tag{16}$$

For low ζ-potentials compared to the thermal voltage $k_B T/e$, the term $\frac{z\,e\,\psi}{k_B T}$ in Eqn. (16) becomes smaller than unity and the *Debye-Hückel approximation* (DHA) can be applied. At room temperature this is fulfilled for $\zeta \ll \frac{25.7\,\text{mV}}{z}$ [80]. In this case the approximation $\sinh(x) \approx x$ is accurate, up to a small error of order $\mathcal{O}(x^3)$ (cf. Taylor's expansion). With this linearization, the symmetric Poisson-Boltzmann equation simplifies to the DHA

$$-\Delta\psi = -\frac{2\, e^2\, z^2\, n_\infty}{\varepsilon_r \varepsilon_0\, k_B\, T}\, \psi = -\kappa^2\, \psi. \tag{17}$$

This equation was originally derived by Debye & Hückel [81] for strong electrolytes [80]. The parameter $\kappa = 1/\lambda_D$, defined by

$$\kappa := \sqrt{\frac{\varepsilon_r \varepsilon_0\, k_B\, T}{2\, e^2\, z^2\, n_\infty}}, \tag{18}$$

is commonly referred to as *Debye-Hückel parameter*. Moreover, the charge density in the fluid is then given by

$$\rho_e(\psi) = -\kappa^2 \varepsilon_e \psi. \tag{19}$$

3.2. Analytical Solutions for Electrophoresis

In this article, we consider spherical particles with uniform ζ-potential distribution as depicted in Fig. 2. The electric potential ψ for such a particle of radius R is represented by the Debye-Hückel equation in
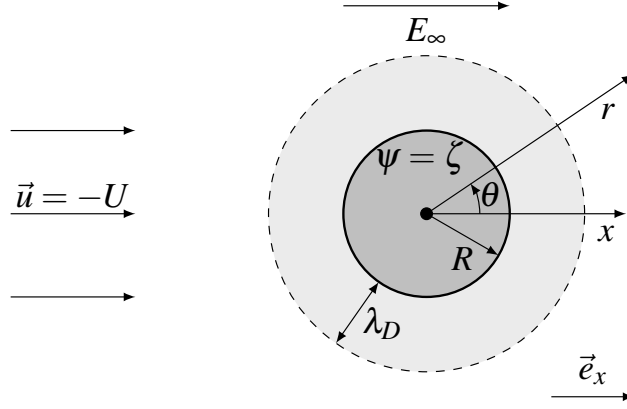

Figure 2: Electrophoresis setup of a stationary (negatively) charged sphere of radius R, surrounded by a double layer and subject to an applied electric field in opposing fluid flow. Similar to [16].

spherical-polar coordinates as

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}\psi}{\mathrm{d}r}\right) = \kappa^2\psi, \tag{20}$$

with radial distance r from the sphere center and subject to the Dirichlet BCs

$$\begin{aligned}\psi &= \zeta &\text{at} \quad r &= R,\\ \psi &\to 0 &\text{as} \quad r &\to \infty.\end{aligned} \tag{21}$$

Solving this equation subject to these BCs results in the electric potential distribution in the surrounding EDL (and beyond) as [80]

$$\psi(r) = \zeta\,\frac{R}{r}e^{-\kappa(r-R)} \qquad \text{for } r \geq R. \tag{22}$$

An analytical solution for the terminal electrophoretic speed U of a single spherical particle at steady state and arbitrary EDL thickness has been derived by Henry [82]. Initially the problem was formulated to account for finite conductivities of particle and medium, and the potential in the EDL was described by the Poisson-Boltzmann equation. The final results, however, were provided for insulating spheres, and sufficiently low ζ-potentials for the Debye-Hückel approximation to hold. Therefore the Debye-Hückel approximation is employed in the following to represent the ion distribution around the particles under electrophoretic motion.

Henry considered a stationary sphere in a steadily moving liquid. The terminal particle speed was imposed as opposing velocity $-U$ far from the particle to bring the system to rest. As shown in Fig. 2, a spherical-polar coordinate system fixed at the particle center with radial distance r and polar angle θ was used. Under the assumption that the electric potential in the EDL is not distorted from its equilibrium distribution by a weak applied field and the fluid flow, the potentials φ and ψ were linearly superimposed. Therefore, the electric potential ψ in the diffuse double layer is described by the Poisson-Boltzmann equation and the applied potential φ by a Laplace equation. The BCs for the Laplace equation applied by Henry represent the insulating particle by homogeneous Neumann BCs at the particle surface and impose the applied field by the inhomogeneous Neumann condition $\partial\varphi/\partial x\big|_{r\to\infty} = -E_\infty$. For the Poisson-Boltzmann equation, the ζ-potential at the hydrodynamic radius R and the decaying potential were imposed as given in Eqn. (21).

Making use of the equations for the electric potential, the Stokes equations with body force term on the right-hand side

$$-\mu_f \, \Delta \, u + \nabla p \;=\; -\rho_e \, \nabla \, (\varphi + \psi) \tag{23}$$
$$\nabla \cdot \vec{u} \;=\; 0, \tag{24}$$

were solved by Henry [82]. In addition to the BC imposing the opposing velocity the no-slip condition $\vec{u}\big|_{r=R} = 0$ was applied at the particle surface.

From the flow field around the particle, the force acting on the particle was obtained by integrating the normal stresses over the sphere surface. To the resulting force that comprises Stokes drag and electric components, the electrostatic force on the particle due to its fixed surface charge was added. The total force must vanish at steady motion and was thus equated with zero, resulting in the electrophoretic velocity

$$\vec{U}_{\mathrm{EP}} = \frac{\varepsilon_e}{\mu_f} \underbrace{\left[\left(\psi_R + R^3 \left(5R^2 \int\limits_\infty^R \frac{\psi}{r^6} \mathrm{d}r - 2 \int\limits_\infty^R \frac{\psi}{r^4} \mathrm{d}r \right) \right) \right]}_{= \zeta \, f(\kappa R)} \vec{E}_{\mathrm{ext}} \tag{25}$$

obtained by Henry for an insulating particle subject to an applied field of strength $\vec{E}_{\mathrm{ext}}$. The function $f(\kappa R)$ introduced in [82] is usually referred to as Henry's function. In Ohshima [83] the following expression is derived, which approximates the integral equations as

$$f(\kappa R) = \frac{2}{3} \left(1 + \frac{1}{2 \left[1 + \frac{2.5}{\kappa R \left(1 + 2 e^{-\kappa R} \right)} \right]^3} \right), \tag{26}$$

with a relative error below 1% for all values of κR [83].

With this approximation, Henry's analytical solution for the electrophoretic velocity of a spherical, non-conducting particle in an unbounded electrolyte solution of dynamic viscosity μ_f and Debye-Hückel parameter κ results as

$$\vec{U}_{\mathrm{EP}} = \frac{2\varepsilon_e \zeta}{3\mu_f} \left(1 + \frac{1}{2 \left[1 + \frac{2.5}{\kappa R (1 + 2e^{-\kappa R})} \right]^3} \right) \vec{E}_{\mathrm{ext}}. \tag{27}$$

This solution is correct to the first order of the ζ-potential, since the relaxation effect is neglected [84].

For the simulations in this article, the particle charge must be known to compute the electrostatic force on the particle. Analytical solutions for electrophoretic motion such as Henry's equation, however, are typically given in terms of the ζ-potential, which is defined at the slip surface between the compact and diffuse EDL layer. Since the particle charge is acquired as a surface charge, for a given ζ-potential the surface charge enclosed by the slip surface is needed in the simulations. The surface charge density is hereby obtained from the overall surface charge bound at the fluid-particle interface and in the Stern layer. This approach is justified by the fact that the electric potential at the Stern surface and the ζ-potential can in general be assumed to be identical [85].

The relation of the surface density σ_s to the ζ-potential is obtained from the Neumann BC on the surface of the insulating particle [84]

$$\sigma_s = -\varepsilon_e \frac{d\psi}{d\vec{r}} \bigg|_{r=R} \tag{28}$$

in case these electrical properties do not vary in angular direction. This condition holds for insignificant permittivity of the insulating particle compared to the fluid permittivity ε_e. With the spatial distribution of ψ around the spherical particle according to Eqn. (22) the $\zeta - \sigma_s$ relationship follows from Eqn. (28) as

$$\sigma_s = \frac{q_s}{4\pi R^2} = \zeta \, \varepsilon_e \left(\frac{1 + \kappa R}{R} \right). \tag{29}$$

11

For a spherical particle with an EDL potential ψ described by the spherical symmetric Poisson-Boltzmann equation, the more general ζ–σ_s relationship

$$\sigma_s = \frac{2\varepsilon_e \kappa k_B T}{ze} \sinh\left(\frac{ze\zeta}{2k_B T} \sqrt{1 + \frac{1}{\kappa R} \frac{2}{\cosh^2\left(\frac{ze\zeta}{4k_B T}\right)} + \frac{1}{(\kappa R)^2} \frac{8\ln\left(\cosh\left(\frac{ze\zeta}{4k_B T}\right)\right)}{\sinh^2\left(\frac{ze\zeta}{2k_B T}\right)}}\right) \tag{30}$$

for 1-1 and 2-1 electrolyte solutions is derived in [86]. The relative error of this approximation w.r.t. the exact numerical results computed by [87] is below 1% for $0.5 \leq \kappa R < \infty$ [84]. Since the latter relationship is derived for the spherical Poisson-Boltzmann equation, it is more general than the relationship (29) for the Debye-Hückel approximation. Therefore Eqn. (30) is applied in Secs. 6.4 and 6.3 to compute the particle charge for a given ζ-potential.

4. Numerical Modeling

4.1. Lattice Boltzmann Method with Forcing

The LBM is a mesoscopic method for the numerical simulation of fluid dynamics based on kinetic theory of gases. This method statistically describes the dynamics of ensembles of fluid molecules in terms of particle distribution functions (PDFs) that represent the spatial and velocity distribution of molecules in *phase space* over time. The temporal and spatial variation of PDFs, balanced by molecular collisions, is described by the Boltzmann equation. The solution of this equation converges towards the Maxwell-Boltzmann velocity distribution of molecules in local thermodynamic equilibrium. For small deviations from this equilibrium, the Navier-Stokes equations can be derived from the Boltzmann equation by means of a Chapman-Enskog expansion (cf. [88, 89]).

In the LBM, the phase space is discretized into a Cartesian lattice $\Omega_{dx} \subset \mathbb{R}^D$ of dimension D with spacing dx, and a set of Q discrete velocities $\vec{c}_q \in \mathbb{R}^D, q \in \{1, \ldots, Q\}$. Moreover, time is discretized as $T_{dt} = \{t_n : n = 0, 1, 2, \ldots\} \subset \mathbb{R}_0^+$, with a time increment of $dt = t_{n+1} - t_n$. The velocities $\vec{c}_q$ are chosen such that within a time increment, molecules can move to adjacent lattice sites or stay at a site. Associated with each of these velocities is a PDF $f_q : \Omega_{dx} \times T_{dt} \mapsto \mathbb{R}$. A forward-difference discretization in time and an upwind discretization in space [90] result in the discrete lattice Boltzmann equation

$$f_q(\vec{x}_i + \vec{c}_q dt, t_n + dt) - f_q(\vec{x}_i, t_n) = dt\mathcal{C}_q + dtF_q, \tag{31}$$

with lattice site positions $\vec{x}_i$, discrete collision operator $dt\mathcal{C}_q$, and discrete body-force term F_q. This equation describes the advection of PDFs between neighboring lattice sites and subsequent collisions.

In general, the collision operator can be represented in terms of the collision matrix S as $dt\mathcal{C}_q = \sum_j S_{qj}\left(f_j - f_j^{eq}\right)$ (cf. [91]), with the vector $\vec{f} := (f_0, f_1, \ldots, f_{Q-1})^\top$ of the PDFs f_q, and $\vec{f}^{eq}$ of the equilibrium distributions f_q^{eq}. The latter are obtained from a low Mach number expansion of the Maxwell-Boltzmann distribution [88]. With a representation of the macroscopic fluid density as $\rho_f = \rho_0 + \delta\rho$ in terms of a reference density ρ_0 and a density fluctuation $\delta\rho$, the equilibrium distribution function for the incompressible LBM is derived in [92] as

$$f_q^{eq}(\rho_0, \vec{u}) = w_q\left[\rho_f + \rho_0\left(\frac{(\vec{c}_q \cdot \vec{u})}{c_s^2} + \frac{(\vec{c}_q \cdot \vec{u})^2}{2c_s^4} - \frac{(\vec{u} \cdot \vec{u})}{2c_s^2}\right)\right], \tag{32}$$

where '$\cdot$' denotes the standard Euclidean scalar product. This distribution function depends on ρ_f, the macroscopic fluid velocity $\vec{u}$, and lattice-model dependent weights w_q. At each instant of time, ρ_f and $\vec{u}$ are given by moments of PDFs as

$$\rho_f(\vec{x}_i, t) = \sum_q f_q(\vec{x}_i, t),$$
$$\vec{u}(\vec{x}_i, t) = \frac{1}{\rho_0}\sum_q \vec{c}_q f_q(\vec{x}_i, t). \tag{33}$$

Moreover, the pressure p is given as $p(\vec{x}_i, t) = c_s^2 \rho_f(\vec{x}_i, t)$ according to the equation of state for an ideal gas. We employ the D3Q19 model of [93] with thermodynamic speed of sound $c_s = c/\sqrt{3}$ for the lattice velocity $c = \mathrm{dx}/\mathrm{dt}$. For this model, the weights w_q are: $w_1 = 1/3$, $w_{2,\ldots,7} = 1/18$, and $w_{8,\ldots,19} = 1/36$. As discussed in [92], f_q^{eq} recovers the incompressible Navier-Stokes equation with at least second-order accuracy of the Mach number $\mathrm{Ma} := \frac{|\vec{u}|}{c_s}$, $\mathcal{O}(\mathrm{Ma}^2)$. In LBM simulations, the kinematic fluid viscosity ν_f is generally determined by a dimensionless relaxation time τ of the collision operator as

$$\nu = \left(\tau - \frac{1}{2}\right) c_s^2 \mathrm{dt}. \tag{34}$$

As shown in [94], with this definition the LBM is second-order accurate in space and time.
Among the different collision operators available for the LBM, we employ the two-relaxation-time (TRT) collision operator of [29, 30]

$$\sum_j S_{qj} \left(f_j - f_j^{\mathrm{eq}}\right) = \lambda_e \left(f_q^e - f_q^{\mathrm{eq},e}\right) + \lambda_o \left(f_q^o - f_q^{\mathrm{eq},o}\right), \tag{35}$$

with the relaxation parameters λ_e and λ_o for even- and odd-order non-conserved moments, respectively. Alternative collision operators either have disadvantages regarding stability and accuracy [95] such as the BGK model [96] or are computationally more costly such as the MRT operator [91] or the cumulant operator [97]. To ensure stability, both relaxation parameters should be within the interval $]-2, 0[$, cf. [98, 30]. The even relaxation parameter is related to the dimensionless relaxation time by $\lambda_e = -\frac{1}{\tau}$ and therefore determines the fluid viscosity. The free parameter λ_o is set to $\lambda_o = -8(2 - 1/\tau)/(8 - 1/\tau)$ in this article, which prevents τ-dependent boundary locations for no-slip BCs as they arise for the BGK operator. Instead, walls aligned with the lattice dimensions are fixed half-way between two lattice sites, as shown in [99]. For the TRT operator, the PDFs are decomposed as $f_q = f_q^e + f_q^o$ into their even and odd components

$$\begin{array}{llll} f_q^e = \frac{1}{2}(f_q + f_{\bar{q}}) & \text{and} & f_q^{\mathrm{eq},e} = \frac{1}{2}(f_q^{\mathrm{eq}} + f_{\bar{q}}^{\mathrm{eq}}) \\ f_q^o = \frac{1}{2}(f_q - f_{\bar{q}}) & \text{and} & f_q^{\mathrm{eq},o} = \frac{1}{2}(f_q^{\mathrm{eq}} - f_{\bar{q}}^{\mathrm{eq}}), \end{array} \tag{36}$$

with $\vec{c}_q = -\vec{c}_{\bar{q}}$. The local equilibrium distribution function in Eqn. (32) is then given by

$$\begin{array}{l} f_q^{\mathrm{eq},e} = w_q \left(\rho_f - \frac{\rho_0}{2c_s^2}(\vec{u} \cdot \vec{u}) + \frac{\rho_0}{2c_s^4}(\vec{c}_q \cdot \vec{u})^2\right) \\ f_q^{\mathrm{eq},o} = w_q \frac{\rho_0}{c_s^2}(\vec{c}_q \cdot \vec{u}). \end{array} \tag{37}$$

At each time step $t_n \in T_{\mathrm{dt}}$ the lattice Boltzmann method performs a *collide–* and a *stream step*

$$\begin{array}{ll} \tilde{f}_q(\vec{x}_i, t_n) = f_q(\vec{x}_i, t_n) & +\lambda_e[f_q^e(\vec{x}_i, t_n) - f_q^{\mathrm{eq},e}(\vec{x}_i, t_n)] \\ & +\lambda_o[f_q^o(\vec{x}_i, t_n) - f_q^{\mathrm{eq},o}(\vec{x}_i, t_n)] \end{array} \tag{38}$$

$$f_q(\vec{x}_i + \vec{e}_q, t_n + \mathrm{dt}) = \tilde{f}_q(\vec{x}_i, t_n) + \mathrm{dt}F_q, \tag{39}$$

where $\tilde{f}_q$ denotes the post-collision state and $\vec{e}_q = \vec{c}_q \mathrm{dt}$ a discrete lattice direction.

In the stream step, the product of dt and the forcing term F_q is added to the post-collision PDFs. The term F_q considers the external effect of body forces acting on the fluid. In this article, the discrete forcing term according to Luo [100] is employed as

$$F_q = w_q \left[\frac{(\vec{c}_q - \vec{u})}{c_s^2} + \frac{(\vec{c}_q \cdot \vec{u})}{c_s^4}\vec{c}_q\right] \cdot \vec{f}_b, \tag{40}$$

with $\vec{f}_b$ representing the electrical body force per unit volume. The forcing terms lead to an additional term in the continuity equation that arises for spatially varying external forces, as shown in [101, 102]. This additional term can be removed by incorporating the external force in the momentum density definition as

$$\vec{u} = \frac{1}{\rho_0} \left(\sum_q f_q \vec{c}_q + \frac{\mathrm{dt}}{2}\vec{f}_b\right). \tag{41}$$

Thus, [101] suggest to use the forcing term (40) in combination with the modified momentum density definition in Eqn. (41) for the BGK. We therefore use this forcing term with second-order accuracy, together with the modified momentum density for the resulting velocity $\vec{u}$.

To increase the computational efficiency of the implementation, the compute-intensive collide step and the memory-intensive stream step are fused to a *stream-collide* step. In the simulations presented in this article, no-slip and free-slip BCs are applied as described in [32].

4.2. Momentum Exchange Approach

To model the fluid-particle interaction and the resulting hydrodynamic interactions of the particles, the momentum exchange approach suggested by [27] is employed. The implementation of this approach in WALBERLA has recently been applied to simulate fluid-particle interactions also at Reynolds numbers beyond the Stokes regime, as presented in [103, 104, 105].

For the momentum exchange approach, the solid particles are mapped onto the lattice by considering each cell whose center is overlapped by a particle a moving obstacle cell. All other lattice cells are fluid cells, resulting in a staircase approximation of the particle surfaces. These surfaces are represented by surface cells indicated by subscript s in the following. On the fluid cells denoted by subscript F, the LBM is applied. To model the momentum transfer from the particles to the fluid, the velocity bounce-back BC

$$f_{\bar{q}}\left(\vec{x}_F, t_n + \mathrm{dt}\right) = \tilde{f}_q\left(\vec{x}_F, t_n\right) - 2\frac{\omega_q}{c_s^2}\rho_0\vec{c}_q \cdot \vec{u}_s \tag{42}$$

is applied at fluid cells with position $\vec{x}_F = \vec{x}_s + \vec{e}_{\bar{q}}$ adjacent to a surface cell at $\vec{x}_s$. This boundary condition introduced in [26] matches the fluid velocity to the local particle surface velocity $\vec{u}_s$.

From the sum of all force contributions due to the momentum transfer from fluid cells to neighboring surface cells (cf. [32]), the overall hydrodynamic force on the particle can be obtained according to [27] as

$$\vec{F}_h = \sum_s \sum_{q \in D_s} \left[2\tilde{f}_q\left(\vec{x}_F, t_n\right) - 2\frac{\omega_q}{c_s^2}\rho_0\vec{c}_q \cdot \vec{u}_s \right] \vec{c}_q \frac{\mathrm{dx}^3}{\mathrm{dt}}. \tag{43}$$

Here, D_s is the set of direction indices q in which a given particle surface cell s is accessed from an adjacent fluid cell. Analogously, the overall torque $\vec{M}_h$ is given by substituting $\vec{c}_q \times (\vec{x}_s - \vec{x}_C)$ for the last $\vec{c}_q$ term in Eqn. (43), with $\vec{x}_C$ representing the particle's center of mass.

The mapping of the solid particles onto the lattice results in fluid cells appearing and disappearing as the particles move. Therefore, at uncovered lattice sites the PDFs must be reconstructed. We set the PDFs at those fluid cells to the equilibrium distribution $f^{\mathrm{eq}}\left(\rho_0, \vec{u}_s(\vec{x}_s(t_n - \mathrm{dt}))\right)$ according to Eqn. (32) dependent on the local particle surface velocity from the previous time step.

4.3. Finite Volume Discretization for Electric Potential Equations

To solve the Debye-Hückel approximation a cell-centered finite volume scheme is applied on the Cartesian lattice Ω_{dx} introduced in Sec. 4.1. Associated with this lattice of spacing dx that represents the computational domain $\Omega \subset \mathbb{R}^3$ is the three-dimensional cell-centered grid G_{dx} defined (cf. [106]) as

$$G_{\mathrm{dx}} := \left\{ \vec{x}_i \in \Omega \;\middle|\; \vec{x}_i = \begin{pmatrix} j - 1/2 \\ k - 1/2 \\ m - 1/2 \end{pmatrix} \mathrm{dx} \;\middle|\; i \in \bigwedge_i, \; (j,k,m) \in \bigwedge_J \right\}, \tag{44}$$

with $\Omega_{\mathrm{dx}} = \Omega \cap G_{\mathrm{dx}}$. For indexing of lattice cells by tuples (j,k,m) of cell indices in the three spatial dimensions, the index set $\bigwedge_J := \{(j,k,m) \mid j = 1,\dots,l_{\mathrm{x}}; \; k = 1,\dots,l_{\mathrm{y}}; \; m = 1,\dots,l_{\mathrm{z}}\}$ is introduced. Here l_{x}, l_{y}, and l_{z} represent the numbers of cells in x, y, and z-direction, respectively. The set $\bigwedge_J$ is related to the set of single cell indices $\bigwedge_i := \{i \mid i = 1,\dots,l_{\mathrm{x}} \cdot l_{\mathrm{y}} \cdot l_{\mathrm{z}}\}$ used for the LBM by a bijective mapping $g : \bigwedge_i \to \bigwedge_J$, according to Eqn. (44).

The finite volume discretization of the Debye-Hückel approximation Eqn. (17) includes volume integration over each lattice cell $\Omega_i = \Omega_{klm} := [\vec{x}_{j-1,k,m}, \vec{x}_{j,k,m}] \times [\vec{x}_{j,k-1,m}, \vec{x}_{j,k,m}] \times [\vec{x}_{j,k,m-1}, \vec{x}_{j,k,m}]$ and applying the divergence theorem to the Laplace operator $\Delta = \nabla \cdot \nabla$, resulting in

$$- \oint_{\partial\Omega_i} \nabla\psi(\vec{x}) \, \mathrm{d}\vec{\Gamma}_i + \kappa^2 \int_{\Omega_i} \psi(\vec{x}) \, \mathrm{d}\vec{x} = 0 \quad \forall \, \Omega_i \in \Omega_{\mathrm{dx}}. \tag{45}$$

Here, $\partial\Omega_i$ denotes the closed surface of the cell, and $\vec{\Gamma}_i$ is a surface directed element. The cell surface consists of six planar faces with constant outward unit normal vectors $\vec{n}_{i_q}$ ($q = 1, \ldots, 6$). Therefore, the surface integral can be decomposed into a sum of integrals [107] as

$$- \oint_{\partial\Omega_i} \nabla\psi(\vec{x}) \, \mathrm{d}\vec{\Gamma}_i = -\sum_{q=1}^{6} \int_{\partial\Omega_{i_q}} \nabla\psi(\vec{x}) \cdot \vec{n}_{i_q} \, \mathrm{d}\Gamma_{i_q}, \tag{46}$$

where q represents LBM direction indices introduced in Sec. 4.1, and $\partial\Omega_{i_q}$ is the common face with the neighboring cell in direction q. The gradients $\nabla\psi(\vec{x}_i) \cdot \vec{n}_{i_q}$ in normal direction of the faces $\partial\Omega_{i_q}$ are approximated at the face centers by central differences of ψ from a neighboring and the current cell as

$$\nabla\psi(\vec{x}_i) \cdot \vec{n}_{i_q}\big|_{\vec{x}_i + \frac{1}{2}\vec{e}_q} \approx \frac{\psi(\vec{x}_i + \vec{e}_q) - \psi(\vec{x}_i)}{\mathrm{dx}}. \tag{47}$$

Here, $\vec{e}_q$ represents the corresponding lattice direction introduced in Sec. 4.1.
Substituting the approximation of the normal fluxes across the surfaces of area $\mathrm{d}\Gamma_{i_q} = \mathrm{dx}^2$, Eqn. (47) into Eqn. (46) results in

$$- \oint_{\partial\Omega_i} \nabla\psi(\vec{x}) \, \mathrm{d}\vec{\Gamma}_i \approx -\sum_{q=1}^{6} \frac{\psi(\vec{x}_i + \vec{e}_q) - \psi(\vec{x}_i)}{\mathrm{dx}} \, \mathrm{dx}^2. \tag{48}$$

Applying the above finite volume discretization to the linear term of the Debye-Hückel approximation results in

$$\kappa^2 \int_{\Omega_i} \psi(\vec{x}) \, \mathrm{d}\vec{x} \approx \kappa^2 \, \psi(\vec{x}_i) \, \mathrm{dx}^3. \tag{49}$$

This additional term enters the central element of the resulting seven-point stencil $\Xi_{\mathrm{dx}}^{\mathrm{DHA}}$ as

$$\frac{1}{\mathrm{dx}^2} \left[\begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 6 + \kappa^2\,\mathrm{dx}^2 & -1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right], \tag{50}$$

and the right-hand side for each unknown is zero.

4.4. Parametrization for Electrokinetic Flows

Numerical simulations are typically performed in terms of dimensionless parameters. To ensure consistently good numerical accuracy independent of the simulated scales, the physical quantities are mapped to a computationally reasonable numerical value range. In LBM simulations usually the quantities are expressed in lattice units. Therefore, physical values must be converted to lattice values before the simulation and vice versa to obtain physical values from simulation results. In the following the lattice unit system employed in this article is presented, providing a common parameterization also for further (electrokinetic) flow scenarios [75].

Physical simulation parameters are given in terms of the *international system of quantities* (ISQ) associated to the *international system of units* (SI [108]) [109]. The SI system comprises the *base units* displayed in Tab. 1, together with the corresponding mutually independent *base quantities* length, time, mass, electric

Table 1: Physical and LBM base quantities for electrokinetic simulations.

kind of quantity	length	time	mass	electr.	thermodyn.	chem.	photometr.
ISQ quantity	x	t	m	I	T	n	I_v
SI unit	m	s	kg	A	K	mol	cd
LBM quantity	x	t	ρ	Φ	–	–	–
lattice unit	dx	dt	ρ_0	V	–	–	–

current, thermodynamic temperature, amount of substance, and luminous intensity. Moreover, *derived units* such as $N = \frac{kg\,m}{s^2}$ are defined as products of powers of base units [108].

For LBM simulations the base quantities are length, time, and mass density. The corresponding *lattice base units* are the (physical) lattice spacing dx, time increment dt, and fluid reference density ρ_0. Therefore, the numerical values of these quantities in *lattice units* (LUs) become unity, as shown in Tab. 2. The lattice parameters representing these dimensionless numerical values in LUs are indicated with subscript L in the following, e.g. dx_L for the lattice spacing. Performing LBM computations on such normalized lattice parameters saves numerical operations: Additional scaling factors are avoided, e.g. in the LBM stream-collide step, when computing the equilibrium distribution function given in Eqn. (32) (cf. $c_L = dx_L/dt_L = 1$) or the macroscopic velocity according to Eqn. (33) (cf. $\rho_{0,L} = 1$).

For the electrokinetic simulations, the electric potential Φ is chosen as electric base quantity. This quantity, however, is not scaled on the lattice but keeps its numerical value that typically lies in a range not too far from unity. In contrast to the SI system, the LU system for electrokinetic simulations requires no base units corresponding to the thermodynamic temperature T or the amount of substance n:

In the simulations, temperature appears only in combination with the Boltzmann constant as energy, i.e., $\mathcal{E} = k_B\,T$, with the derived unit $[\mathcal{E}] = [m]\,[x]^2/[t]^2$. With the relation of mass and mass density $[m] = [\rho_0]\,[x]^3$, this unit can be represented in lattice base units (see Tab. 1).

Moreover, by representing the molar concentration with unit $[c_i] = \frac{mol}{l}$ in terms of the number density as $n_i = c_i\,N_A$ with Avogadro's number $N_A = 6.022\,14 \cdot 10^{23}\,\frac{1}{mol}$, the unit 'mol' cancels out, yielding $[n_i] = [x]^{-3}$.

With the choice of the potential Φ as base quantity, the electric current I becomes a derived quantity in the LU system. The unit of I can be derived from the energy in electrical units $[\mathcal{E}] = [I]\,[\Phi]\,[t]$ and the above definition of energy in terms of lattice units $[\mathcal{E}] = [\rho_0]\,[x]^5/[t]^2$. Equating both relations results in the derived lattice unit

$$[I] = \frac{[\rho_0]\,[x]^5}{[\Phi]\,[t]^3}. \tag{51}$$

In Tab. 2 different physical quantities and their SI units are displayed, as well as their representation by (dimensionless) lattice parameters and their lattice units. The conversion of physical quantities to lattice units requires their division by powers of the LBM base quantities with the corresponding SI units. Since the potential Φ has the same numerical value in physical and lattice units, the LBM base unit of the potential is simply '1 V'. Therefore, the derived lattice unit 'Ampere' for the electric current is given by $A = \frac{\rho_0\,dx^5}{V\,dt^3}$ (see Eqn. (51)). The corresponding scale factors for converting the physical parameters (e.g. ν) to the associated lattice parameters (e.g. ν_L) are shown in Tab. 2. Multiplication with the inverse scale factors converts the lattice parameters back to physical quantities.

5. Extension of waLBerla for Electrophoresis Simulations

Electrophoresis simulations require the mutual coupling of fluid dynamics, rigid body dynamics, and electro-statics, as shown in Fig. 1. In addition to electrostatic and hydrodynamic interactions, the applied field acts on the EDL charge and thereby affects fluid flow and particle motion. For the electrophoresis algorithm presented below, the equilibrium description of the EDL potential in terms of the linear Debye-Hückel equation is employed. Therefore, the predominant retardation effect is recovered in the simulations.

Table 2: Relation of physical quantities and lattice parameters for electrokinetic simulations.

physical quantity	SI unit	lattice parameter	numerical value	lattice unit
dx (lattice spacing)	m	dx_L	$\frac{1}{dx} dx \ (= 1)$	dx
dt (time increment)	s	dt_L	$\frac{1}{dt} dt \ (= 1)$	dt
ρ_0 (fluid density)	$\frac{kg}{m^3}$	$\rho_{0,L}$	$\frac{1}{\rho_0} \rho_0 \ (= 1)$	ρ_0
Φ (electr. potential)	V	Φ_L	$\frac{1}{V} \Phi$	V
L (length)	m	L_L	$\frac{1}{dx} L$	dx
ν (kinem. viscosity)	$\frac{m^2}{s}$	ν_L	$\frac{dt}{dx^2} \nu$	$\frac{dx^2}{dt}$
$\vec{u}$ (velocity)	$\frac{m}{s}$	$\vec{u}_L$	$\frac{dt}{dx} \vec{u}$	$\frac{dx}{dt}$
m (mass)	kg	m_L	$\frac{1}{\rho_0 dx^3} m$	$\rho_0 dx^3$
$\vec{F}$ (force)	$\frac{kg\,m}{s^2}$	$\vec{F}_L$	$\frac{dt^2}{\rho_0 dx^4} \vec{F}$	$\frac{\rho_0 dx^4}{dt^2}$
I (electr. current)	A	I_L	$\frac{V\,dt^3}{\rho_0 dx^5} I$	$\frac{\rho_0 dx^5}{dt^3 V}$
e (elem. charge)	A s	e_L	$\frac{V\,dt^2}{\rho_0 dx^5} e$	$\frac{\rho_0 dx^5}{dt^2 V}$
ε_0 (vac. permittivity)	$\frac{A\,s}{V\,m}$	$\varepsilon_{0,L}$	$\frac{V\,dt^2}{\rho_0 dx^5} \varepsilon_0$	$\frac{\rho_0 dx^5}{V\,dt^2}$
$\vec{E}$ (electr. field)	$\frac{V}{m}$	$\vec{E}_L$	$\frac{dx}{V} \vec{E}$	$\frac{V}{dx}$
$\mathcal{E}$ (energy)	J	$\mathcal{E}_L$	$\frac{dt^2}{\rho_0 dx^5} \mathcal{E}$	$\frac{\rho_0 dx^5}{dt^2}$

The applied potential φ and the EDL potential ψ are linearly superimposed (cf. Henry's equation, Sec. 3), which is valid for weak applied fields when the EDL distortion by the field is negligible. Thus, the applied electric field can be imposed directly, without solving the associated Laplace equation. In the following the main concepts of the WALBERLA framework are described, together with the functionality for electrophoresis simulations implemented therein.

5.1. Design Concepts of WALBERLA

WALBERLA is a framework for massively parallel multiphysics simulations with a MPI-based distributed memory parallelization that is specifically designed for supercomputers. The main software design goals of this framework are flexibility to combine models of different effects, extensibility to allow the incorporation of further effects and details, e. g., for electrokinetic flows, and generality to support further applications [75]. These goals are reached by integrating the coupled simulation models into WALBERLA in a modular fashion that avoids unnecessary dependencies between the modules. This way, the modules can be augmented by more sophisticated models or models tailored to a certain application, and functionality from different modules can be combined flexibly. The modular code structure also provides excellent maintainability, since modifications of the code in one module do not affect other modules. Developers can therefore efficiently locate faulty modules and find bugs inside these modules, also by systematically utilizing automatic tests.

In addition to *modules*, the WALBERLA code structure comprises a *core* for sequence control that initializes data structures, performs the time stepping, and finalizes the simulation on each parallel process. By means of *applications*, multiphysics simulations can be defined by assembling the associated functionality and coupled models from the modules. The coupling strategy for multiphysics simulations is based on accessing mutually dependent data structures (see [31] for more details). These data strucutres are defined in the modules that implement models for the different physical effects. Also infrastructural and utility functionality is encapsulated in modules, e. g., for domain setup, MPI communication, BC handling, parameterization, or simulation data output. For parallel simulations the discretized simulation domain is decomposed into equally sized *blocks* of cells that can be assigned to different parallel processes. In this *block concept*, each block contains a layer of surrounding ghost cells that is needed for BC treatment and parallelization. For parallelization, cell data of neighboring processes is copied to the ghost layer, dependent on

the data dependencies of the unknowns located on a given block. Moreover, metadata of a block specifies its location in the simulation domain or its rank for MPI communication. The *communication concept* provides a simple and flexible communication mechanism tailored to simulations on Cartesian grids and facilitates various communication patterns. Individual work steps of a simulation algorithm are specified as *sweeps* that are executed on a block-parallel level. The *sweep concept* defines a structure in which callable objects (i.e. kernels) implemented in the modules can be specified at compile time. By means of dynamic application switches, specific kernels tailored to a given computer architecture or implementing a desired model variant, can be selected at run time. For time-dependent simulations, the sweeps are organized in a *timeloop* that specifies the order in which the individual work steps are executed at each time step. To facilitate iterative solvers, sweeps can also be nested to repeatedly perform a grid traversal until a termination criterion is met.

The *boundary condition concept* for handling multiple physical fields, numerical methods, and governing equations, introduced in [31], is applied in this article for moving obstacles with electric BCs. This concept relies on *flags* to indicate for each boundary cell the kind of boundary treatment that to is be performed, with an individual *BC* flag for each condition. Moreover, cells adjacent to a boundary are indicated with a *nearBC* flag and non-boundary cells with a *nonBC* flag. Individual sets of these flags are defined for each governing equation. Due to specific LBM requirements, the boundary handling is performed such that the BCs are fulfilled when a boundary cell is accessed from a *nearBC* cell in the susequent sweep. The abstract boundary handling is implemented in the *bc* module and provides functionality for all BCs are handled either as direct or direction-dependent treatment. In direction-dependent treatment the BC value is set at a boundary cell dependent on the value at a neighboring cell, whereas in direct BC treatment the BC value is directly set at a boundary cell [31]. The actual boundary handling functionality is implemented in corresponding BC classes whose functions are executed when the associated *BC* flag is found.

The *parameterization concept* for multiphysics simulations introduced in [75] is based on the conversion of physical parameters to lattice units before the simulation, as described in Sec. 4.4. This approach ensures consistent parameters in all WALBERLA modules, independent of the underlying physics. Individual modules can therefore be developed independently w.r.t. the common lattice unit system.

Simulation parameters and BCs are typically provided to WALBERLA through an input file. To ensure a consistent parameter set and a correct mapping of the physical quantities to lattice units, the class *PhysicalCheck* has been introduced in WALBERLA in [110]. This class checks the simulation parameter set specified in the input file for completeness and physical validity and converts the parameters to lattice units. Since the quantities are converted based on the SI system, the unit Ampere is re-defined for PhysicalCheck according to Eqn. (51) to support simulations including electric effects.

5.2. Overview of WALBERLA Modules for Electrophoresis Simulations

In the following an overview of the modules relevant for electrophoresis simulations is given. For fluid simulations with the LBM, the *lbm* module implements various kernels for the stream-collide step with the different collision operators and forcing terms described in Sec. 4.1. The classes for treating the corresponding BCs are provided by the associated *lbm_bc* module. In the *lbm* module block-fields of cells are provided for the PDFs, the velocity, the density, and an external force. The external force field is used for coupling the LBM to other methods, e.g. via the forces exerted by electric fields on the EDL and the fluid in electrophoresis simulations. The PDF and velocity field are accessed by *moving_obstacle* module functions for the simulation of moving particles.

The *moving_obstacle* module facilitates simulations of fluid-particle interactions with the momentum exchange method by implementing kernels for moving obstacle sweeps and providing the corresponding data structures. This module furthermore provides setup functions for initializing and connecting the *pe* to WALBERLA. For the moving obstacle handling, an obstacle-cell relation field is provided that stores for each lattice cell overlapped by a *pe* object a pointer to this object. Moreover, from the *lbm* module the PDF source field is accessed for the hydrodynamic force computation and the reconstruction of PDFs. In the *lbm* velocity field, body velocities are stored and accessed in the moving boundary treatment of the LBM.

For the computation of the electric potential distribution, the *lse_solver* module described in Sec. 5.3 is employed. This module has been implemented for solving large sparse linear systems of equations as they

arise from the discretization of the electric potential equations (see Sec. 4.3). The corresponding BC classes are implemented in the *pot_bc* module described in Sec. 5.4. The data structures defined in the *lse_solver* module, accessed by the application and other modules, include the stencil field representing the system matrix as well as the scalar fields for the solution and for the RHS.

The *electrokin_flow* module was designed for facilitating electrokinetic flow simulations. This module provides kernels for coupling the involved methods as well as the setup and parameterization of simulations such as electrophoresis. The setup includes initializing the stencils and RHS from the *lse_solver* module according to the finite volume discretization of the Debye-Hückel equation presented in Sec. 4.3. The kernels for imposing the electric potential BCs on the moving particles and for computing the electrostatic forces on fluid and particles are described in Sec. 5.5 and Sec. 5.6, respectively. Finally, the coupled algorithm for electrophoresis provided by the *electrokin_flow* module is presented in Sec. 5.7.

5.3. Solver for Linear Elliptic PDEs with Moving Boundaries

The solver module *lse_solver* in waLBerla for large linear systems of equations has been designed as an efficient and robust black-box solver on block-structured grids [111, 31, 75]. To specify the problem to be solved, the application sets up the system matrix, right-hand side, and BC handling.

In the *lse_solver* module, solver sweeps are pre-defined that perform the iterations at the position where they are added to the timeloop. For a given simulation setup, the employed solver is selected via the input file where also the solver parameters and BCs are specified. An iterative solver requires a nested sweep that is executed until a specific convergence criterion is satisfied. For all implemented solvers, these sweeps share a common structure. This structure is displayed for the SOR sweep `solveTimeVaryingBCSOR` for moving boundaries and linear PDEs such as the Debye-Hückel approximation in Fig. 3 as an activity diagram. For
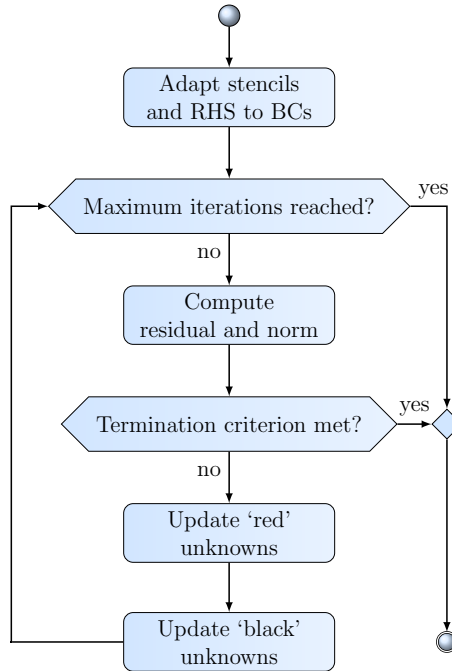

Figure 3: Activity diagram for SOR sweep `solveTimeVaryingBCSOR` with time varying boundary conditions (BCs), e.g., due to moving particles

the employed discretization of the electric potential equations, the system matrix is represented by D3Q7 stencils. These stencils comprise an entry for each, the center and the six cardinal directions.

In the setup function for this SOR sweep, the communication for the ghost layer exchange of the solution field in the initialization phase is set up first. Then the SOR solver sweep is added to the timeloop, and the

kernels for relaxation, communication, and BC treatment are specified as solver sub-sweeps. For parallel execution, the SOR algorithm is implemented in red-black order.

The filled circle at the top of the diagram in Fig. 3 indicates the starting point of the sweep in the timeloop. At the beginning of the sweep `solveTimeVaryingBCSOR` for moving boundaries, the stencils are constructed to incorporate the BCs according to the present boundary locations. Furthermore, the RHS is adapted to these BCs before the solver iterations start. For this purpose, a sub-sweep with a kernel for re-setting the stencils and RHS is executed before the iteration loop, followed by the BC treatment functions `adaptStencilsBC` and `adaptRHSBC` described in Sec. 5.4. Then the standard parallel Red-Black SOR sweep is performed until the termination criterion is met. This sweep comprises a sub-sweep for computing the residual and its L_2-norm for the termination criterion, as well as two solver sub-sweeps for the SOR update of the 'red' and the 'black' unknowns, respectively. In these sub-sweeps, the quasi-constant stencil optimization technique introduced in [31] is employed. Based on the residual L_2-norm, the termination criterion for the simulations performed in this article is provided as residual reduction factor RES_{RF} w.r.t. the initial norm of the simulation.

For multigrid solvers as applied in [31] to charged particle simulations in absence of ions in the fluid, the red-black update sub-sweeps in Fig. 3 are replaced by solver sub-sweeps of a V-cycle. To apply the SOR sweep to varying stencils of linearized PDEs, such as the symmetric Poisson-Boltzmann equation, only the sub-sweep for adapting the stencils and RHS to the BCs is performed at the beginning of each iteration instead of once before the iterations start (see [75]).

5.4. Boundary Condition Handling for Solver Module

The implicit BC handling used and initiated by the solver module has been introduced in [31]. This boundary handling is based on incorporating the BCs into the stencils and right-hand side of the finite volume discretization. That way, at an iterative update of a near-boundary value, the method implicitly uses the new values for the BCs. For Dirichlet boundaries, the boundary values are linearly extrapolated to the boundary cell and for Neumann BCs the boundary values are approximated by central differences. For both the stencils and the right-hand side, a direction-dependent BC treatment is used.

The functions for this BC treatment are implemented in the *pot_bc* module. This module employs own *nonBC* and *nearBC* flags for the BC handling of scalar potentials. Moreover, for each BC class in this module an associated *BC* flag is defined. For the employed cell-centered discretization, the module contains one class for each, Neumann and Dirichlet BCs.

For incorporating the BCs into the stencils the kernel `adaptStencilsBC` is implemented. This kernel iterates over all lattice cells to find scalar potential *nearBC* cells. At each cell with *nearBC* flag, the kernel employs the D3Q7 stencil directions to iterate over the neighboring cells. In directions of a cell with scalar potential *BC* flag, the stencil entry of the *nearBC* cell, associated with the direction of the *BC* flag, is adapted accordingly.

The function `adaptRHSBC` employs the standard boundary handling of the *bc* module to invoke the *pot_bc* kernels for adapting the RHS to the BCs. To this end, the direction-dependent BC treatment kernels in the corresponding BC classes implement the RHS adaption depending on the BC value. The BC value is specified in the input file for static BCs, or in a previous time step for BCs of moving particles (see Sec. 5.5). To facilitate such complex boundaries, the BC classes store the BC values and the corresponding boundary cell ranges. The latter are stored in a memory-efficient way either as cell intervals or as cell sets. Moreover, to allow the computation of scalar potential gradients directly from the solution field, the BC values are set in this field at boundary cells when the RHS is adapted. From these BC values and from the solution at the *nearBC* cell, the value at the boundary cell required for the gradient can be extrapolated.

5.5. Electric Potential Boundary Condition Handling for Moving Particles

Prior to the EDL potential distribution computation by the *lse_solver* module, uniform ζ-potentials or surface charge densities are imposed at the moving particles by means of scalar potential BCs. These electrical surface properties are specified in the input file for different particle types with a common *uid* (unique identifier) defined in the *pe*. To this end, the sweep function `setPotBC_ChargParticles` is implemented

in the *electrokin_flow* module that maps the charged particles onto the lattice and sets the electric potential BC values and the associated BC handling flags at the corresponding cells. The function first overwrites the values of all cells in the RHS field with zero to remove the values from the previous BC treatment. Then the mapping is performed for all movable rigid bodies located in the subdomain of each process. For each particle the mapping is conducted in an extended axis-aligned bounding box that surrounds this rigid body and the associated *nearBC* flags. The mapping is realized in three steps:

1) First, all scalar potential *nearBC* and *BC* flags from the previous time step are removed, and the scalar potential *nonBC* flags are set. Moreover, the previous BC values and the associated cells in the BC class instances are removed.

2) Then, for particles with prescribed electric BCs, the BC handling for the *lse_solver* module (see Sec. 5.4) is prepared. For each rigid body with a *uid* for which a surface property is specified in the input file, the associated BC is obtained. The cells overlapped by this particle are gathered and are added together with the BC value to the corresponding BC class instance. Moreover, the *BC* flag is set at the overlapped cells.

3) Finally, the *nearBC* flag is set at all cells adjacent to a *BC* cell.

Each step is performed for all bodies on a process before the next step begins, to prevent that for particles with overlapping bounding boxes the flags from a previous step are overwritten.

5.6. Computing Electrostatic Forces on Fluid and Particles

The electric forces acting on the ions in the fluid are incorporated into the incompressible Navier-Stokes equation by the body force term $\vec{f}_b = -\rho_e(\psi)\,\nabla\,(\varphi + \psi)$, which coincides with the corresponding term employed in Henry's solution (see Sec. 3). Due to the linear superposition of the electric potential components, the gradient is applied to both components separately. Since the applied field $\vec{E}_{\text{ext}} = -\nabla\varphi$ is given, only the EDL potential gradient must be computed.

For the computation of the electric field due to the EDL, the *electrokin_flow* module provides a kernel that performs the gradient computation at all scalar potential *nonBC* cells. The gradient of the electric potential is computed, as previously introduced in [31], by means of finite differences that provide $\mathcal{O}(\mathrm{dx}^2)$ accuracy. Where possible, an isotropy-preserving D3Q19 stencil is used (cf. [112]) instead of a D3Q7 stencil. With the LBM D3Q19 stencil, the gradient can be computed using w_q-weighted differences of neighboring values in 18 directions $\vec{e}_q$ as

$$\nabla\psi(\vec{x}_b) \approx \frac{1}{w_1}\sum_{q=2}^{19} w_q\,\psi(\vec{x}_b + \vec{e}_q) \cdot \frac{\vec{e}_q}{\mathrm{dx}^2}. \tag{52}$$

At *nearBC* cells the D3Q7 stencil is applied to compute the gradient of ψ from the BC values stored at particle and boundary cells in the BC treatment (see Sec. 5.4). The obtained electric field is stored in a field of cells that is accessed in the body force computation.

For the computation of the electric body force and of the electrostatic force exerted on the particles, a further kernel is implemented in the *electrokin_flow* module:
The kernel first iterates on each parallel process over all lattice cells to compute the body force at scalar potential *nonBC* cells. This force is computed as product of charge density $\rho_e(\psi)$ and total electric field $\vec{E}_{\text{total}} = \vec{E}_{\text{ext}} - \nabla\psi$. The relation of charge density and EDL potential follows Eqn. (19). The obtained electric body force is written to the external force field of the *lbm* module that is accessed by the LBM kernels with forcing. Then the kernel iterates over all non-fixed particles residing on the current parallel process to compute the electrostatic force. For each of these particles the force is computed from the particle charge and the applied field as $\vec{F}_C = q_e\,\vec{E}_{\text{ext}}$ and is then added to the particle.

5.7. Algorithm for Electrophoresis

The overall parallel algorithm for electrophoresis simulations with WALBERLA is shown in Alg. 1. After the setup and initialization phase, the electric BCs for the EDL potential are set at the moving charged particles by means of `setPotBC_ChargParticles` at each time step. Then the Debye-Hückel approximation is

---

Algorithm 1: Electrophoresis Algorithm

---

foreach *time step,* **do**
    // solve Debye-Hückel approximation (DHA):
    set electric BCs of particles
    while *residual ≥ tol* **do**
        apply SOR iteration to DHA
    // couple potential solver and LBM:
    begin
        compute electric field due to EDL
        compute charge density in fluid
        compute electric body force
        apply electrostatic force to particles
    // solve lattice Boltzmann equation with forcing, considering particle velocities:
    set velocity BCs of particles
    begin
        perform fused stream-collide step
    // couple potential solver and LBM to *pe*:
    begin
        apply hydrodynamic force to particles
        pe moves particles depending on forces

---

solved by means of the SOR sweep `solveTimeVaryingBCSOR`. The iterations are performed until the specified termination criterion for the residual is met.

From the obtained EDL potential distribution and the applied field the electric body force exerted on the fluid is computed, as described in Sec. 5.6. First the kernel computing the electric field caused by the EDL is applied. Then the kernel for the electric body force computation from the charge density distribution in the fluid and the total electric field is invoked. This kernel additionally applies the electrostatic force exerted by the applied field to the particles.

Then the rigid body mapping sweep described in Sec. 4.2 is performed, imposing the particle velocities for the subsequent LBM sweep. In that parallel sweep, an LBM kernel with fused stream-collide step and forcing term (see Sec. 4.1) is employed to compute the fluid motion influenced by the moving particles and by the electrostatic force exerted on ions in the EDL.

After the LBM sweep, the hydrodynamic forces on the particles are computed by the momentum exchange method. The obtained hydrodynamic force contributions and the electrostatic forces are then aggregated by the *pe*. From the resulting forces and torques, the new particle velocities and positions are computed in the subsequent *pe* simulation step by the PFFD algorithm [113] that additionally resolves rigid body collisions.

6. Electrophoresis Simulations

In the following, the correctness of the electrophoresis algorithm is validated and its parallel performance is analyzed. The electric potential computation is validated for a sphere with uniform surface charge surrounded by an EDL. This sphere is placed in a micro-channel subject to an applied electric field. Moreover the flow field caused by the electrophoretic motion of the sphere in the micro-channel is visualized, together with the electric potential and the surrounding ions to qualitatively show the correctness of the simulations. Then the electrophoretic velocity and the retardation by the counter-ions in the EDL is validated w.r.t. Henry's solution for different sphere sizes and values of κR. Finally, weak scaling experiments are presented and the parallel performance and scalability of the electrophoresis algorithm is shown up to more than four million spheres.

6.1. Simulation Setups

In all simulations, the insulating spherical particles are suspended in a symmetric aqueous electrolyte solution with the physical parameters listed in Tab. 3. Moreover, gravitational are effects neglected to ensure

Table 3: Parameters of electrolyte solution and particles used in all simulations. Shown are kinematic viscosity ν_f, density ρ_f, permittivity ε_e, ion valence z, temperature T, and particle density ρ_p.

ν_f	ρ_f	ε_e	z	T	ρ_p
$1.00 \cdot 10^{-6}\ \frac{\mathrm{m}^2}{\mathrm{s}}$	$1000\ \frac{\mathrm{kg}}{\mathrm{m}^3}$	$78.54 \cdot \varepsilon_0$	1	$293\,\mathrm{K}$	$1195\ \frac{\mathrm{kg}}{\mathrm{m}^3}$

that the particle motion is driven solely by electric forces. The EDL thickness λ_D is in the order of the particle diameter in all simulations and always greater than the 12 lattice sites required for a sufficient resolution, as observed in the electro-osmotic flow simulations in [75]. A uniform surface-potential distribution is chosen with a ζ-potential sufficiently low to approximate the Poisson-Boltzmann equation by the Debye-Hückel approximation. For this approximation analytical solutions are known for the potential distribution and for the electrophoretic velocity. The linear system of equations resulting from the finite volume discretization of the Debye-Hückel equation is solved by the SOR method that is sufficient for the quickly decaying electric potential due to the counter-ions in the EDL. For the SOR, a relaxation parameter of $\omega_{\mathrm{SOR}} = 1.7$ is applied in all simulations. The LBM is employed with TRT operator and second-order forcing term by Luo and re-defined momentum density (see Sec. 4.1).

For the validation simulations the setup with a single sphere is depicted in Fig. 4. The sphere is placed on
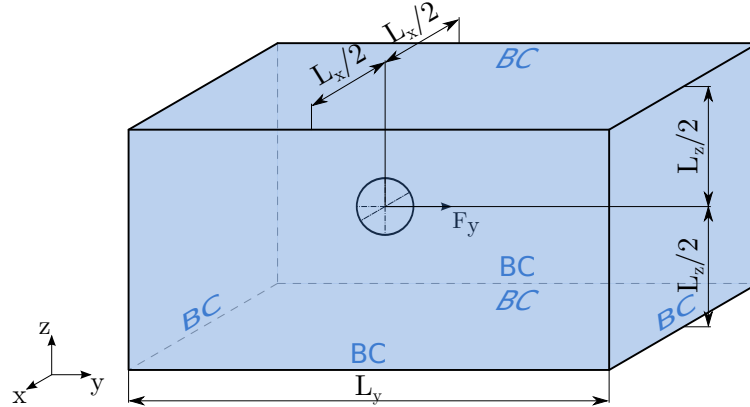

Figure 4: Setup for electrophoresis of spheres in square duct with different BCs.

the longitudinal axis of a cuboid domain of size $L_x \times L_y \times L_z$ at an initial position of y_0, and an electrostatic force in y-direction acts on the sphere.

For the EDL potential validation in Sec. 6.2 the parameters in the left part of Tab. 4 are used. The simulation domain is discretized with the lattice spacing dx and at the surface of the sphere of radius R_L the ζ-potential is imposed. In addition to the parameters in Tab. 3 the bulk ion concentration c^∞ is simulated to obtain the displayed values of the Debye-Hückel parameter κ and of the characteristic EDL thickness λ_D.

For the electrophoretic motion visualization in Sec. 6.3 additionally the parameters in the right part of Tab. 4 are employed. The sphere's charge q_s is obtained from the surface charge density σ_s according to the ζ–σ_s relationship (30), multiplied by the surface area of the sphere, as $q_s = 4\pi R^2 \sigma_s$ [80]. The high electric field magnitude E_y is chosen to keep the number of simulation time steps at a minimum. From the applied LBM relaxation parameter τ the time increment dt results according to Eqn. (34) from the chosen viscosity ν_f and dx.

Table 4: Simulation parameters for validation of EDL potential (left) and electrophoresis of sphere in micro-channel (whole table).

ζ	c^∞	κ	q_s	E_y
$-10.0\,\mathrm{mV}$	$5.00 \cdot 10^{-6}\,\frac{\mathrm{mol}}{\mathrm{l}}$	$7.41 \cdot 10^6\,\frac{1}{\mathrm{m}}$	$-19.9 \cdot 10^{-18}\,\mathrm{A\,s}$	$-4.7 \cdot 10^6\,\frac{\mathrm{V}}{\mathrm{m}}$
dx	R_L	$\lambda_{D,L}$	τ	dt
$10 \cdot 10^{-9}\,\mathrm{m}$	12	13.49	6.5	$200 \cdot 10^{-12}\,\mathrm{s}$

The velocity validation experiments in Sec. 6.4 and scaling experiments in Sec. 6.5 are performed on SuperMUC[1] of the Leibniz Supercomputing Centre LRZ in Garching (Germany). For these experiments the parameters in Tab. 5 are used. The ζ-potential has the same absolute value as in the electrophoretic motion simulations in Sec. 6.3. From the value of c^∞ and from the electrolyte parameters in Tab. 3 results the value of κ and a Debye length λ_D of approximately 15 lattice sites. The LBM is employed with the relaxation time $\tau = 6$, yielding the time increment dt. As in Sec. 6.3, a high electric field E_y is chosen

Table 5: Simulation parameters for electrophoretic velocity validation w.r.t. Henry's solution and for scaling experiments. Parameters with subscript 'Sc' are applied in scaling runs.

ζ	c^∞	κ	E_y	$E_{y,\mathrm{Sc}}$
$10.0\,\mathrm{mV}$	$1.60 \cdot 10^{-5}\,\frac{\mathrm{mol}}{\mathrm{l}}$	$13.3 \cdot 10^6\,\frac{1}{\mathrm{m}}$	$99.0 \cdot 10^6\,\frac{\mathrm{V}}{\mathrm{m}}$	$38.0 \cdot 10^6\,\frac{\mathrm{V}}{\mathrm{m}}$
dx	$\lambda_{D,L}$	τ	dt	$R_{L,\mathrm{Sc}}$
$5.00 \cdot 10^{-9}\,\mathrm{m}$	15.08	6	$45.8 \cdot 10^{-12}\,\mathrm{s}$	6

for the validation to keep the number of simulation time steps low. In the scaling experiments the electric field is reduced to a more realistic magnitude of $E_{y,\mathrm{Sc}}$ and spheres of radius $R_{L,\mathrm{Sc}}$ are simulated. To obtain different values of κR, the sphere radii in Tab. 6 are used in the validation experiments. For the sphere sizes

Table 6: Surface charges q_s used in velocity validation (and scaling) experiments.

R_L	4	6	8	9	12
$q_s/(10^{-18}\,\mathrm{A\,s})$	2.21	3.67	5.36	6.29	9.43

and parameters used in the scaling and validation experiments, the associated surface charges are displayed in Tab. 6. These charges are again obtained from the general ζ–σ_s relationship (30). For all parameters in Tab. 5, the maximum relative deviation of σ_s for the general relationship from the exact value for the Debye-Hückel approximation is below 0.2%.

6.2. Electric Potential in an EDL around a Sphere

To validate the computation of the EDL potential ψ around a charged particle, a sphere is simulated in a large domain of size $128\,\mathrm{dx} \times 256\,\mathrm{dx} \times 128\,\mathrm{dx}$. The analytical solution of the Debye-Hückel equation (20) representing the EDL potential around the sphere is given by Eqn. (22). For the validation, a spherical particle of radius $R_L = 12$ with initial position $y_0 = 64\,\mathrm{dx}$ is chosen.

Despite its quick decay, the analytical solution of the electric potential at the domain boundaries differs from zero. Thus, the values of ψ at these boundaries are set to the analytical solution by means of Dirichlet conditions. To solve the Debye-Hückel equation subject to these BCs, a residual reduction factor of $\text{RES}_{\text{RF}} = 2 \cdot 10^{-7}$ is employed as termination criterion for the SOR method.

The analytical (ψ) and numerical (ψ^*) solution at the initial particle position are depicted in Fig. 5 along
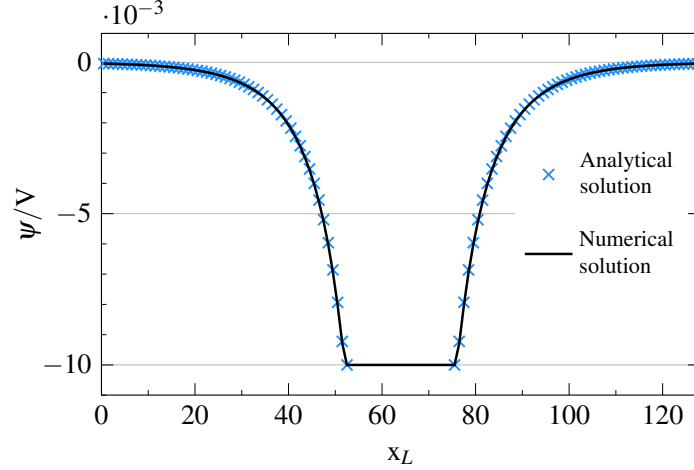
Figure 5: Analytical and numerical solution for EDL potential of sphere with uniform surface charge.

a line in x-direction through the sphere center. Both graphs agree very well, showing the correctness of the finite volume discretization and the applied SOR solver, as well as the boundary handling at the particle surface. Inside the insulating particle, the electric potential is not computed explicitly, due to the uniform surface potential and the resulting symmetric distribution of ψ in the sphere. The electrostatic force needed to compute the particle motion is computed directly from the applied field and the particle charge, instead of the electric potential gradient as in [31].
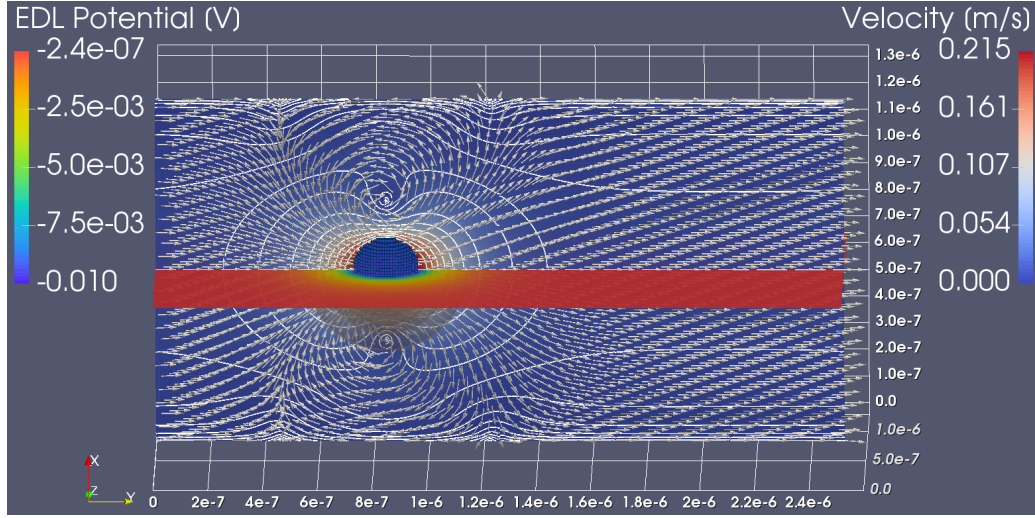
6.3. Electrophoresis of a Sphere in a Micro-Channel

The application of the electric field in Tab. 4 to the micro-channel setup described in Sec. 6.1 gives rise to an electrophoretic motion of the sphere. Due to the applied field and the resulting electrostatic force of $F_{\text{C,y}} = 933 \cdot 10^{-12}\,\text{N}$ the particle moves in y-direction, retarded by the channel walls and the opposing force on the EDL. For the chosen parameters, the terminal particle speed of $U_{\text{EP}} = 224.5\,\frac{\text{mm}}{\text{s}}$ (or $U_{\text{EP},L} = 4.49 \cdot 10^{-3}$) is obtained for free space according to Henry's solution (27), corresponding to a particle Reynolds number of $\text{Re}_{p,d} = 0.054$. Since gravitational effects are neglected, the particle density in Tab. 3 only has an impact on the time required to reach steady-state.

In the simulation, periodic BCs are applied in y-direction of the domain of size $128\,\text{dx} \times 256\,\text{dx} \times 128\,\text{dx}$. At all other walls, no-slip conditions are applied for the LBM and homogeneous Neumann conditions for the electric potential. At each time step, the Debye-Hückel equation is solved by SOR with the termination criterion $\text{RES}_{\text{RF}} = 2 \cdot 10^{-7}$.

In Fig. 6, the results of the electrophoresis simulation are visualized at different time steps. The EDL potential ψ in the y-z plane through the domain center is displayed, together with semi-transparent hemispherical equipotential surfaces representing the excess counter-ions in the EDL. The flow field around the moving sphere is visualized in the x-z plane through the domain center. Arrows of uniform length indicate the flow direction, while the velocity magnitude is represented by the shown color-scale and by twelve white isosurface contour lines with logarithmic intervals in the range of $13.67 \cdot 10^{-3}\,\frac{\text{m}}{\text{s}}$ to $6.08 \cdot 10^{-6}\,\frac{\text{m}}{\text{s}}$.

The flow field around the moving particle shown in Fig. 6(a) is fully developed after 5001 time steps, and the particle has already attained its terminal velocity. Due to the periodicity of the domain, a channel flow in axial direction has emerged from the particle motion, as indicated by the contour lines. Moreover, a vortex

(a) Results after 5001 time steps.
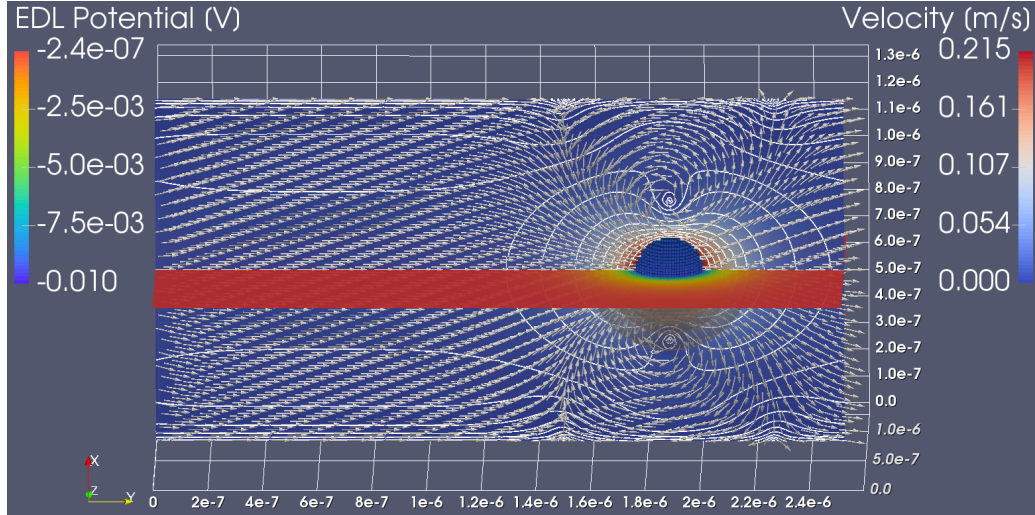
(b) Results after 30 001 time steps.

Figure 6: Electrophoresis of spherical particle in micro-channel with insulating no-slip walls. Visualization of flow field in x-y plane, EDL potential in y-z plane, and ion charge distribution (as equipotential surfaces) around charged particle.

has formed between the sphere and the surrounding no-slip walls. The flow field moves with the particle that translates along the channel centerline (see Fig. 6(b)). Because of the equilibrium representation of the EDL, the distribution of ψ is at all time steps symmetric w.r.t. the sphere center, almost up to the boundary.

6.4. Validation of Electrophoretic Motion of a Sphere

To quantitatively validate the overall electrophoresis algorithm, the electrophoretic velocity of spheres with uniform surface charge is compared to Henry's solution (27) for a spherical particle in an unbounded electrolyte solution. In the simulation experiments, a sphere with the ζ-potential in Tab. 5 is moving under the influence of an applied electric field in a large domain filled with an electrolyte solution. The simulations are performed for different values of κR by varying the sphere radius while keeping the EDL thickness constant. For the different sphere radii, the forces F_C on the particle resulting from the applied field in Tab. 5 and the associated surface charges, are displayed in Tab. 7. For validation, the relative deviation

$\Delta_r U$ of the obtained terminal sphere velocity from the theoretical velocity in Eqn. (27) is evaluated. The employed simulation parameters given in Sec. 6.1 are chosen such that the electrophoretic motion is in the Stokes regime and the EDL potential is governed by the Debye-Hückel approximation. Therefore, the superposition principle is assumed to hold. Thus, from the obtained relative deviation from the analytical solution in an unbounded domain $\Delta_r U$, the relative deviation due to wall effects and volume mapping errors $\Delta_r U_{\mathrm{Stokes}}$ will be subtracted. These relative deviations were examined in [75] for several domain sizes and sphere radii. For no-slip BCs, the wall effect was shown to comply with analytical and experimental results. In the experiments, domain sizes are used for which the relative deviation from Stokes velocity is close to $\Delta_r U_{\mathrm{Stokes}} = -3\,\%$ for free-slip BCs.

For the different sphere radii and the associated values of κR, the electrophoretic velocities according to Henry's solution are displayed in Tab. 7. These velocities correspond to particle Reynolds numbers $\mathrm{Re}_{p,d}$ from 0.018 to 0.057 for the particle diameters of 40 nm to 120 nm.

To quantify the retardation by the opposing force on the EDL, the variable $\mathrm{EP}_{\mathrm{Ret}} = \frac{U_{\mathrm{EP}} - U_{\mathrm{EM}}}{U_{\mathrm{EM}}}$ is introduced. This variable represents the relative deviation of the electrophoretic velocity of a particle with charge q_s in presence of the electric double layer from the migration velocity U_{EM} of a particle with the same charge in absence of surrounding ions. For the examined sphere radii, this retardation is in the range of 20% to 42%.

Then the domain sizes are listed in Tab. 7, together with the initial positions in movement direction that

Table 7: Electrophoresis parameters and domain sizes dependent on sphere radii R_L. For the relations of sphere radius to EDL thickness κR, electrostatic forces F_C, theoretical electrophoretic velocities U_{EP}, Reynolds numbers $\mathrm{Re}_{p,d}$, and electrophoretic retardation $\mathrm{EP}_{\mathrm{Ret}}$ are given. Listed in lower part are domain sizes per dimension $L_{\mathrm{x,y,z}}$, initial sphere positions y_0, process numbers per dimension $\#\mathrm{proc}_{\mathrm{x,y,z}}$, and relative deviations of sphere velocities in free-slip domain from Stokes velocity $\Delta_r U_{\mathrm{Stokes}}$.

R_L	4	6	8	9	12
κR	0.265	0.398	0.530	0.597	0.796
$F_C/(10^{-12}\,\mathrm{N})$	219	363	530	622	934
$U_{\mathrm{EP}}/(\frac{\mathrm{mm}}{\mathrm{s}})$	461	464	466	468	471
$\mathrm{Re}_{p,d}$	0.018	0.028	0.037	0.042	0.057
$\mathrm{EP}_{\mathrm{Ret}}/\%$	-20.6	-27.7	-33.6	-36.2	-42.8
$L_{\mathrm{x,z}}/\mathrm{dx}$	864	1280	1632	1632	1632
$L_{\mathrm{y}}\ /\mathrm{dx}$	1248	1536	2048	2048	2048
y_0/dx	392	588	784	882	1176
$\#\mathrm{proc}_{\mathrm{x,y,z}}$	$8 \times 16 \times 8$	$8 \times 16 \times 16$	$16 \times 16 \times 32$	$16 \times 16 \times 32$	$16 \times 16 \times 32$
$\Delta_r U_{\mathrm{Stokes}}/\%$	-3.04	-2.65	-2.98	-2.89	-3.00

correspond to $y_0 = 98 \times R_L$.

For the electric potential, homogeneous Neumann BCs are applied at the walls in x- and z-direction as $\frac{\partial \Phi}{\partial \vec{n}}\big|_{\Gamma_W \cup \Gamma_E \cup \Gamma_B \cup \Gamma_T} = 0\,\frac{\mathrm{V}}{\mathrm{m}}$. To improve convergence [31], homogeneous Dirichlet BCs are applied at the walls in y-direction as $\Phi\big|_{\Gamma_N \cup \Gamma_S} = 0\,\mathrm{V}$. The particle is located at sufficient distance from these walls for the EDL potential to decay to approximately zero, and therefore these BCs do not affect the electric potential distribution. As termination criterion of the SOR solver for the Debye-Hückel equation, a residual reduction factor of $\mathrm{RES}_{\mathrm{RF}} = 1 \cdot 10^{-6}$ is used.

The parallel simulations are run on 64 to 512 nodes of SuperMUC with the numbers of processes listed in Tab. 7. Within the execution time of 37 h to 48 h, a number of 70 296 to 74 000 time steps are performed, and the spheres cover a distance of about 300 dx. The high numbers of time steps are chosen to ensure that the spheres reach steady-state motion, and that large numbers of sampling values are available to compute the terminal velocities.

The sphere velocities are sampled every 20 time steps during the simulation. In the second half of the simulation, the terminal particle velocity is reached. Thus, the mean particle velocity U^* and the velocity

fluctuations $\delta_U = \frac{U^*_{\max} - U^*_{\min}}{U^*}$ due to volume mapping effects are computed from the last 50% output values. In the considered range of time steps additionally the number of time steps between two SOR calls and the number of SOR iterations is monitored. The average number of time steps between two SOR calls decreases from $\Delta \mathrm{TS_{SOR}} = 24$ to $\Delta \mathrm{TS_{SOR}} = 3$ as R_L increases from 4 to 12. Likewise, the average number of SOR iterations per solver call decreases from 451 iterations to 198 iterations for the respective sphere radii. The obtained terminal velocity in lattice units U^*_L and the fluctuations are displayed in Tab. 8. As expected,

Table 8: Simulation results of electrophoretic velocity validation for different sphere sizes. Shown are the theoretical velocities $U_{\mathrm{EP},L}$ for unbounded domains in lattice units, obtained velocities U^*_L and fluctuations δ_U, relative deviations $\Delta_\mathrm{r} U$ of U^*_L from $U_{\mathrm{EP},L}$, and relative deviations $\Delta_\mathrm{r} U_{\mathrm{EP}}$ corrected by hydrodynamic wall and mapping effects.

R_L	4	6	8	9	12
$U_{\mathrm{EP},L}/10^{-3}$	4.227	4.249	4.274	4.286	4.320
$U^*_L/10^{-3}$	4.119	4.144	4.120	4.135	4.151
$\delta_U/\%$	2.52	1.52	0.83	0.68	0.29
$\Delta_\mathrm{r} U/\%$	-2.56	-2.48	-3.59	-3.51	-3.91
$\Delta_\mathrm{r} U_{\mathrm{EP}}/\%$	0.5	0.2	-0.6	-0.6	-0.9

the fluctuations decrease with increasing sphere resolution. Moreover, the relative deviation of the obtained velocity U^* from the theoretical electrophoretic velocity U_{EP} given by $\Delta_\mathrm{r} U = (U^* - U_{\mathrm{EP}})/U_{\mathrm{EP}}$ is listed in Tab. 8. For all examined sphere radii, the obtained velocities in the confined domain are by 2.5% to 3.9% lower than the theoretical values of U_{EP} for a particle in an unbounded electrolyte solution. The effect of the confinement on the particle velocity is deducted by subtracting the relative deviation from Stokes velocity $\Delta_\mathrm{r} U_{\mathrm{Stokes}}$ in Tab. 7 for the corresponding domain sizes from the relative electrophoretic velocity deviation $\Delta_\mathrm{r} U$ obtained in the electrophoresis simulation. From the resulting relative deviations $\Delta_\mathrm{r} U_{\mathrm{EP}} = \Delta_\mathrm{r} U - \Delta_\mathrm{r} U_{\mathrm{Stokes}}$, the inaccuracies due to electric effects in the electrophoresis simulation are assessed. As can be seen from the values of $\Delta_\mathrm{r} U_{\mathrm{EP}}$ in Tab. 8, the simulations results agree with the theoretical values with relative deviations below 1%.

6.5. Parallel Performance and Scaling

Following common practice in parallel computing, scaling experiments [114] were performed to assess the parallel efficiency and scalability of the algorithm. The performance experiments presented in the following were conducted on SuperMUC Phase 1. Phase 1 comprises 18 Thin Islands with 512 Thin nodes each, connected by a high speed InfiniBand FDR10 interconnect. Each node contains two Intel Xeon E5-2680 "Sandy Bridge-EP" octa-core processors that were running at 2.5 GHz, and 32 GB DDR3 RAM. The code is built with the GCC 5.4.0 compiler, IBM MPI 1.4, and Boost 1.57.

For the performance measurements, the parameters described in Sec. 6.1 are used. A block of size 144^3 cells with 64 spheres of radius $R = 6\,\mathrm{dx}$ is assigned to each process. The spheres are initially arranged as a set of 4^3 equidistantly placed particles with center-to-center distance $36\,\mathrm{dx}$ and therefore weakly overlapping double layers. Here the center of the first particle is placed at position $x = y = z = 19\,\mathrm{dx}$ w.r.t the coordinate system in Fig. 4.

In all experiments 240 time steps are performed and the simulation domain is periodic in y-direction. At all other walls, no-slip conditions are applied for the LBM and homogeneous Neumann conditions for the electric potential.

For the single-node weak scaling performance evaluation, an increasing number of processes is allocated to the processors until all 16 cores are occupied. Starting from a serial process, the problem size is extended successively in y-direction to $1 \times 4 \times 1$ processes and then from $1 \times 3 \times 2$ to $1 \times 7 \times 2$ processes. The maximum problem size per node with $2 \times 4 \times 2$ processes is kept constant for the weak scaling experiments. In these experiments the problem size is extended successively in all three dimensions in a round-robin fashion. The extension begins in z-direction with $2 \times 4 \times 4$ processes on two nodes, up to $32 \times 64 \times 32$ processes on 4096 nodes. This maximum problem size comprises $196 \cdot 10^9$ lattice cells and $4.19 \cdot 10^6$ particles.

For the Debye-Hückel equation the L_2 norm of the initial residual has a value of $3.37 \cdot 10^{-3}$ at all problem sizes of the weak scaling experiment. In order to permanently reduce this residual by a factor of $\mathrm{RES_{RF}} = 4 \cdot 10^{-6}$, an average number of 29 SOR iterations per time step is required on 128 nodes and above, as observed for 640 time steps. Therefore at each time step 29 SOR iterations are applied to solve the Debye-Hückel equation.

About $1.95\,\%$ of the domain is covered by moving obstacle cells that are not updated by the LBM and the SOR. To quantify the performance, we use the measure *million fluid lattice updates per second* (MFLUPS) [115] as the number of fluid cells that can be updated within one second by the LBM and the SOR. For the SOR this metric indicates the performance to obtain the solution up to a given accuracy, with several iterations per update.

On a single node the total runtime increases from $582\,\mathrm{s}$ on one core to $717\,\mathrm{s}$ on 16 cores. This is equivalent to $81\,\%$ parallel efficiency of the overall algorithm. In Fig. 7 the speedup of the whole algorithm and its main parts is presented for up to 16 MPI processes. Both LBM and SOR exhibit good scaling and
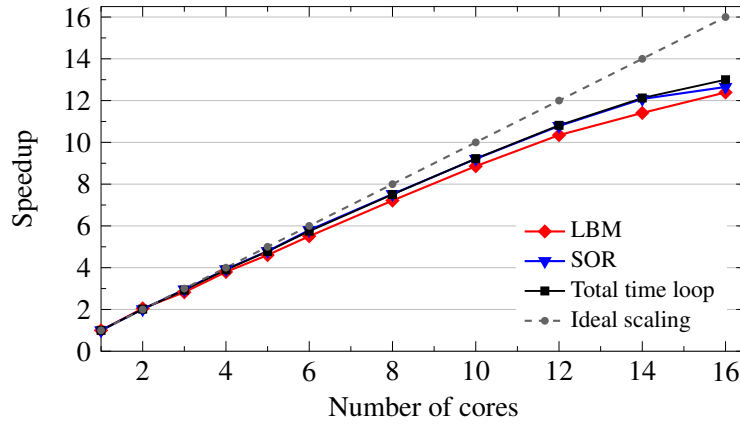

Figure 7: Single-node weak scaling performance of the whole algorithm and SOR and LBM sweeps for 240 time steps on SuperMUC.

achieve $77\,\%$ and $79\,\%$ parallel efficiency on a full node, respectively. Their performance mainly restricted by computation-related data transfers, with less than $2.6\,\%$ communication share between the processes for the LBM. The relative share of intra-node communication on the SOR runtime more than doubles from $3.6\,\%$ on 14 processes to $7.6\,\%$ on 16 processes, due to the additional communication in the third dimension. The single-node performance of the LBM and the SOR sweep is 61.2 MFLUPS and 28.9 MFLUPS, respectively. These values will serve as base performance in the following, where we employ 16 processes per node because this leads to the highest overall performance.

In the weak scaling experiments on up to 4096 nodes on SuperMUC presented in the following for 240 time steps, the problem size is successively doubled as described above. The runtimes of all parts of the algorithm are shown in Fig. 8 for different problem sizes, indicating their shares on the total runtime. This diagram is based on the maximal (for SOR, *pe*) or average (others) runtimes of the different sweeps among all processes. The upper part of the diagram shows the cost of fluid-simulation related sweeps, such as LBM, moving obstacle mapping (MOmap), and hydrodynamic force computation (HydrForce) sweep. In the middle, the cost of the *pe* sweep is shown. Below, the costs of sweeps related to electric effects are displayed. These include computing the electrostatic forces on particles and fluid (LBMForce), computing the electric field in the EDL (EDLField), SOR, and the sweep setting the electric potential BCs at the particles (PotBCmap).

For a more precise evaluation, the exact figures are shown in Tab. 9 for one node and 4096 nodes. The total runtime (Whl) differs from the sum of the individual sweeps, since different sweeps are slow on different processes. Sweeps whose runtimes depend on the problem size—mainly due to increasing MPI
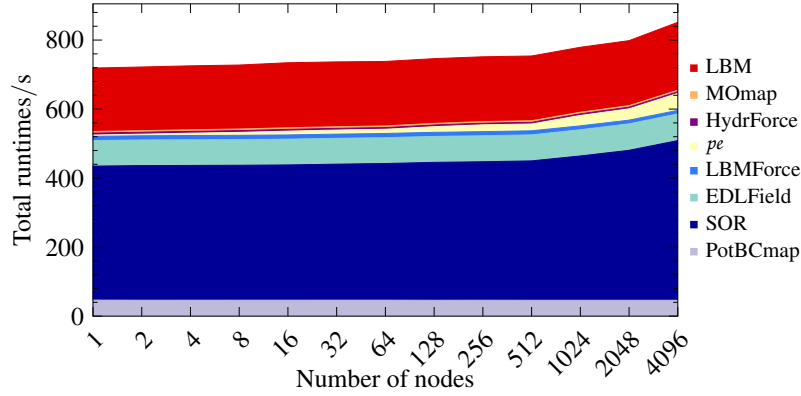
Figure 8: Runtimes of electrophoresis algorithm sweeps for 240 time steps for an increasing number of nodes (16 cores per node).

Table 9: Time of the whole algorithm and its sweeps for 240 time steps on a single node and on 4096 nodes. Relative share of sweeps on total runtime given in parentheses.

#n.	Whl t/s	SOR t/s (%)	LBM t/s (%)	pe t/s (%)	EDLField t/s (%)	Oth t/s
1	717	390 (54)	184 (26)	4 (1)	74 (10)	69
4096	825	474 (56)	197 (24)	49 (6)	77 (9)	66

communication—are LBM, MG, *pe*, and EDLField. Overall, LBM and SOR take up more than 80 % of the total time, w.r.t. the runtimes of the individual sweeps. The sweeps that scale perfectly—MOmap, HydrForce, LBMForce, and PotBCmap—are summarized as 'Oth'. Here PotBCmap has with 6 % the highest portion of the total runtime.

Overall, the multiphysics algorithm achieves 87 % parallel efficiency on 4096 nodes. Since most time
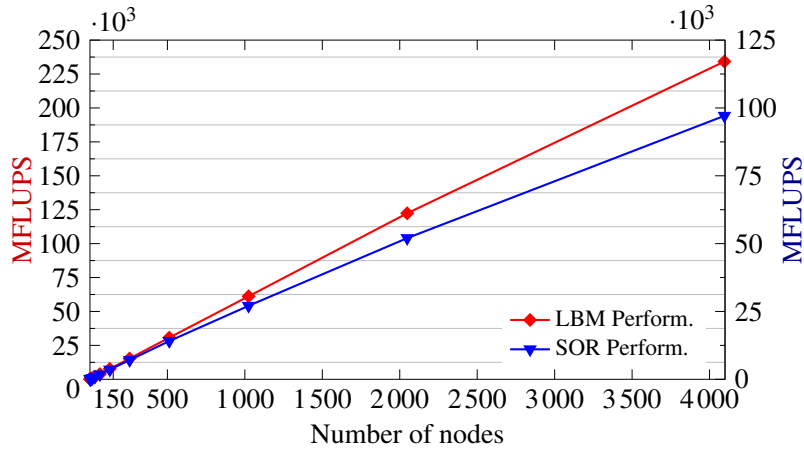

Figure 9: Weak scaling performance of LBM and SOR sweep for 240 time steps on SuperMUC.

is spent to execute LBM and SOR, we will now analyze them in more detail. Fig. 9 displays the parallel performance for different numbers of nodes. On 4096 nodes, SOR executes 97 102 MFLUPS, corresponding to a parallel efficiency of 82 %. The LBM performs 234 227 MFLUPS, with 94% parallel efficiency.

Finally, the average execution times of the different SOR sub-sweeps are presented in Fig. 10. Like in the single-node scaling, the parallel performance of the SOR is degraded mainly by increasing communication
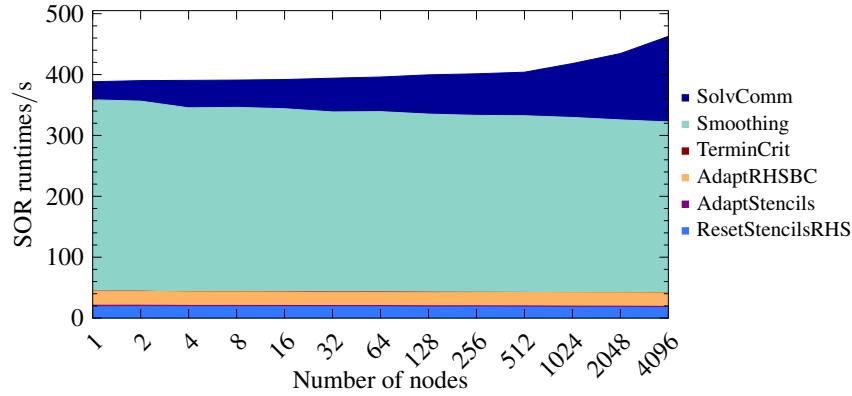
Figure 10: Average runtimes of SOR sweeps for 240 time steps with 29 iterations per time step.

costs. All other sweeps exhibit ideal scaling. Among these sweeps, most time is spent for smoothing, whereas all other parts require only a small portion of the overall SOR time. These are the sweeps for resetting the stencils and right-hand side to their initial values (ResetStencilsRHS, 18 s), for adapting the right-hand side to the BCs (AdaptRHSBC, 22 s), and adapting the stencils to the BCs (AdaptStencils, 2 s). The sweep for checking the termination criterion is negligible (< 0.02 s).

7. Conclusion

In this article, a coupled multiphysics algorithm for parallel simulations of the electrophoretic motion of geometrically resolved particles in electrolyte solutions is presented. The physical effects are simulated by means of the lattice Boltzmann method for fluid dynamics, a physics engine for rigid body dynamics, and a scalar iterative solver for electric potentials. These components are integrated into the parallel software framework waLBerla to simulate the electrophoretic motion of fully resolved charged particles in microfluidic flow. The simulations include fluid-particle and electrostatic particle interactions. Additionally electric effects on ions around the homogeneously charged particles are recovered.

The current work is an extension of [31], where the electrical migration of charged particles without ions in the fluid was validated and excellent parallel performance was shown for more than seven millions of interacting charged particles. In the present article, the opposite net charge in the electric double layer (EDL) around the charged particles due to ions in the fluid is considered, together with its effect on the fluid motion that counteracts particle motion. To this end, the electric potential distribution in the fluid due to the EDLs is computed that causes an electric body force on the fluid. This quasi-equilibrium distribution recovers the motion of ions in the fluid along with the charged particles while neglecting EDL distortion.

The overall electrophoresis algorithm is introduced and an overview of the coupled functionality implemented in the involved waLBerla modules is given. For the simulations, a solver sweep for time-varying boundary conditions has been developed that is presented here for the parallel SOR method employed to solve the EDL potential equation. Based on the multiphysics boundary handling concept [31] an efficient parallel algorithm is implemented to impose electric potential boundary conditions on the moving particles. These methods can also be employed for other governing equations with spatially varying boundary conditions that model physical effects different from electric fields. The presented parallel electrophoresis simulations are also facilitated by a joint parameterization concept for the different coupled governing equations and numerical methods implemented in waLBerla. This concept is based on lattice Boltzmann requirements and is applicable and extensible to further multiphysics simulations.

For the electrophoresis simulations in this article, the electric potential in the double layer is shown to coincide with analytical solutions. The obtained terminal electrophoretic velocities comply with analytical solutions for different proportions of the particle radii to double layer thickness. These validation results verify the correctness of the implementation and the coupling of the different methods. Moreover, the

observed relative errors in the modeling of electric effects are below 1%. The retardation effect caused by the presence of the EDL is shown to be significant for the examined sphere radii, reducing the sphere velocity up to 42%. For the electrophoretic motion in a micro-channel, the flow field and the electric potential distribution are visualized, including the ion charge distribution in the EDL surrounding the particle.

The parallel performance of the algorithm is examined on SuperMUC. The weak scaling of the overall algorithm yields a parallel efficiency of 87% on $65\,536$ cores, simulating more than four million charged particles and their surrounding by EDLs with 1.95% solid volume fraction. The overall parallel performance is based on a carefully designed, extensible software architecture that exceeds the capabilities of alternative simulation software by several orders of magnitude. Most time of the electrophoresis algorithm is spent for the SOR solver and the LBM. The SOR algorithm scales with 82% parallel efficiency, achieving more than $97.1 \cdot 10^9$ fluid cell updates per second to solve the Debye-Hückel approximation. The LBM for fluid-particle interaction scales almost perfectly with 94% parallel efficiency, achieving more than $234 \cdot 10^9$ fluid cell updates per second.

The presented algorithm can be applied to find design parameters in industrial and medical applications, e. g., for optimal separation efficiency of charged biological particles in lab-on-a-chip devices, depending on fluid, particle, and electrolyte properties. Our algorithms were shown to correctly recover fluid-particle interactions for elongated particles in [32]. These future simulations may therefore include suspended, possibly charged particles of various shapes including spheres, spherocylinders, and particles of more complex shapes, e. g., to represent different biological particles. Also pairwise van der Waals forces can be added easily, to facilitate simulations of electrophoretic deposition in material science applications.

The electrophoresis algorithm introduced here is well suited for massively parallel simulations. In the current implementation of this algorithm, the EDL thickness is restricted to values in the order of the particle radius. Therefore, adaptive lattice refinement as in [23] may be employed to allow for thinner double layers relative to the particle size. For the incorporation of transient effects in the simulations including EDLs, the link-flux method implemented into waLBerla in [116] and [41] may be employed. This method was extended in [41] to simulate electrophoresis, enabling the simulation of non-equilibrium ion distributions in the EDL. Due to the higher computational complexity of the link-flux method compared to the equilibrium approach in this article, the maximum number of particles will be lower than in our approach.

Acknowledgements

References

[1] K. J. Ptasinski, P. J. A. M. Kerkhof, Electric Field Driven Separations: Phenomena and Applications, Sep. Sci. Technol. 27 (8-9) (1992) 995–1021. `doi:10.1080/01496399208019021`.

[2] S. Zhang, R. Tan, K. Neoh, C. Tien, Electrofiltration of Aqueous Suspensions, J. Colloid Interface Sci. 228 (2) (2000) 393–404. `doi:10.1006/jcis.2000.6966`.

[3] Y.-H. Weng, K.-C. Li, L. H. Chaung-Hsieh, C. Huang, Removal of humic substances (HS) from water by electro-microfiltration (EMF), Water Res. 40 (9) (2006) 1783–1794. `doi:10.1016/j.watres.2006.02.028`.

[4] A. Mahmoud, J. Olivier, J. Vaxelaire, A. Hoadley, Electrical field: A historical review of its application and contributions in wastewater sludge dewatering, Water Res. 44 (8) (2010) 2381–2407. `doi:10.1016/j.watres.2010.01.033`.

[5] L. Besra, M. Liu, A review on fundamentals and applications of electrophoretic deposition (EPD), Prog. Mater. Sci. 52 (1) (2007) 1 – 61. `doi:10.1016/j.pmatsci.2006.07.001`.

[6] I. Zhitomirsky, Cathodic electrodeposition of ceramic and organoceramic materials. Fundamental aspects, Adv. Colloid Interface Sci. 97 (1–3) (2002) 279–317. `doi:10.1016/S0001-8686(01)00068-9`.

[7] P. Sarkar, D. De, T. Uchikochi, L. Besra, Electrophoretic Deposition (EPD): Fundamentals and Novel Applications in Fabrication of Advanced Ceramic Microstructures, Springer, 2012, Ch. 5, pp. 181–215. `doi:10.1007/978-1-4419-9730-2_5`.

[8] Y. Kang, D. Li, Electrokinetic motion of particles and cells in microchannels, Microfluid Nanofluid. 6 (4) (2009) 431–460. `doi:10.1007/s10404-009-0408-7`.

[9] A. A. S. Bhagat, H. Bow, H. W. Hou, S. J. Tan, J. Han, C. T. Lim, Microfluidics for cell separation, Med. Biol. Eng. Comput. 48 (10) (2010) 999–1014. `doi:10.1007/s11517-010-0611-4`.

[10] D. R. Gossett, W. M. Weaver, A. J. Mach, S. C. Hur, H. T. K. Tse, W. Lee, H. Amini, D. Di Carlo, Label-free cell separation and sorting in microfluidic systems, Anal. Bioanal. Chem. 397 (8) (2010) 3249–3267. `doi:10.1007/s00216-010-3721-9`.

[11] N. Pamme, Continuous flow separations in microfluidic devices, Lab Chip 7 (2007) 1644–1659. `doi:10.1039/B712784G`.

[12] D. G. Hert, C. P. Fredlake, A. E. Barron, Advantages and limitations of next-generation sequencing technologies: A comparison of electrophoresis and non-electrophoresis methods, Electrophoresis 29 (23) (2008) 4618–4626. `doi:10.1002/elps.200800456`.

[13] G. W. Slater, C. Holm, M. V. Chubynsky, H. W. de Haan, A. Dube, K. Grass, O. A. Hickey, C. Kingsburry, D. Sean, T. N. Shendruk, L. Zhan, Modeling the separation of macromolecules: A review of current computer simulation methods, Electrophoresis 30 (5) (2009) 792–818. `doi:10.1002/elps.200800673`.

[14] F. Keller, H. Nirschl, W. Dörfler, E. Woldt, Efficient numerical simulation and optimization in electrophoretic deposition processes, J. Eur. Ceram. Soc. 35 (9) (2015) 2619–2630. `doi:10.1016/j.jeurceramsoc.2015.02.031`.

[15] D. Sheehan, Physical Biochemistry: Principles and Applications, John Wiley & Sons, 2013.

[16] R. F. Probstein, Physicochemical Hydrodynamics: An Introduction, 2nd Edition, Butterworths series in chemical engineering, Butterworth Publishers, 1989.

[17] H. C. Chang, L. Y. Yeo, Electrokinetically Driven Microfluidics and Nanofluidics, Cambridge Univ. Press, 2009.

[18] O. Stern, ZUR THEORIE DER ELEKTROLYTISCHEN DOPPELSCHICHT, Z. Elektrochem. 30 (21-22) (1924) 508–516. `doi:10.1002/bbpc.192400182`.

[19] T. Preclik, U. Rüde, Ultrascale Simulations of non-smooth granular dynamics, Comp. Part. Mech. 2 (2) (2015) 173–196. `doi:10.1007/s40571-015-0047-6`.

[20] K. Iglberger, U. Rüde, Massively Parallel Rigid Body Dynamics Simulation, Comp. Sci. Res. Dev. 23 (3-4) (2009) 159–167. `doi:10.1007/s00450-009-0066-8`.

[21] C. Feichtinger, S. Donath, H. Köstler, J. Götz, U. Rüde, WaLBerla: HPC software design for computational engineering simulations, J. Comput. Sci. 2 (2) (2011) 105–112. `doi:10.1016/j.jocs.2011.01.004`.

[22] C. Godenschwager, F. Schornbaum, M. Bauer, H. Köstler, U. Rüde, A Framework for Hybrid Parallel Flow Simulations with a Trillion Cells in Complex Geometries, in: Proc. Int. Conf. on High Performance Computing, Networking, Storage and Analysis, SC '13, ACM, 2013, pp. 35:1–35:12. `doi:10.1145/2503210.2503273`.

[23] F. Schornbaum, U. Rüde, Massively Parallel Algorithms for the Lattice Boltzmann Method on NonUniform Grids, SIAM J. Sci. Comput. 38 (2) (2016) C96–C126. `arXiv:1508.07982`, `doi:10.1137/15M1035240`.

[24] S. Chen, G. D. Doolen, Lattice Boltzmann Method for Fluid Flows, Annu. Rev. Fluid Mech. 30 (1) (1998) 329–364. `doi:10.1146/annurev.fluid.30.1.329`.

[25] C. K. Aidun, J. R. Clausen, Lattice-Boltzmann method for complex flows, Annu. Rev. Fluid Mech. 42 (1) (2010) 439–472. `doi:10.1146/annurev-fluid-121108-145519`.

[26] A. J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation, J. Fluid Mech. 271 (1994) 285–309. `doi:10.1017/S0022112094001771`.

[27] N. Q. Nguyen, A. J. C. Ladd, Lubrication corrections for lattice-Boltzmann simulations of particle suspensions, Phys. Rev. E 66 (4) (2002) 046708. `doi:10.1103/PhysRevE.66.046708`.

[28] J. Götz, K. Iglberger, M. Stürmer, U. Rüde, Direct Numerical Simulation of Particulate Flows on 294912 Processor Cores, in: Proc. 2010 ACM/IEEE Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis, SC '10, IEEE, 2010, pp. 1–11. `doi:10.1109/SC.2010.20`.

[29] I. Ginzburg, J.-P. Carlier, C. Kao, Lattice Boltzmann approach to Richards' equation, in: W. G. G. C. T. Miller, M. W. Farthing, G. F. Pinder (Eds.), Computational Methods in Water Resources, Vol. 55 of Developments in Water Science, Elsevier, 2004, pp. 583–595. `doi:10.1016/S0167-5648(04)80083-2`.

[30] I. Ginzburg, F. Verhaeghe, D. d'Humières, Two-relaxation-time lattice Boltzmann scheme: About parametrization, velocity, pressure and mixed boundary conditions, Commun. Comput. Phys. 3 (2) (2008) 427–478.

[31] D. Bartuschat, U. Rüde, Parallel Multiphysics Simulations of Charged Particles in Microfluidic Flows, J. Comput. Sci. 8 (2015) 1–19. `doi:10.1016/j.jocs.2015.02.006`.

[32] D. Bartuschat, E. Fischermeier, K. Gustavsson, U. Rüde, Two Computational Models for Simulating the Tumbling Motion of Elongated Particles in Fluids, Comput. Fluids 127 (2016) 17–35. `doi:10.1016/j.compfluid.2015.12.010`.

[33] S. Jomeh, M. Hoorfar, Study of the effect of electric field and electroneutrality on transport of biomolecules in microreactors, Microfluid Nanofluid. 12 (1) (2012) 279–294. `doi:10.1007/s10404-011-0871-9`.

[34] P. Kler, C. Berli, F. Guarnieri, Modeling and high performance simulation of electrophoretic techniques in microfluidic chips, Microfluid Nanofluid. 10 (1) (2011) 187–198. `doi:10.1007/s10404-010-0660-x`.

[35] M. Chau, T. Garcia, P. Spiteri, Asynchronous grid computing for the simulation of the 3D electrophoresis coupled problem, Adv. Eng. Softw. 60–61 (2013) 111–121. `doi:10.1016/j.advengsoft.2012.11.010`.

[36] M. Chau, P. Spiteri, H. C. Boisson, Parallel numerical simulation for the coupled problem of continuous flow electrophoresis, Int. J. Numer. Meth. Fluids 55 (10) (2007) 945–963. `doi:10.1002/fld.1502`.

[37] B. Giera, L. A. Zepeda-Ruiz, A. J. Pascall, J. D. Kuntz, C. M. Spadaccini, T. H. Weisgraber, Mesoscale particle-based model of electrophoresis, J. Electrochem. Soc. 162 (11) (2015) D3030–D3035. `doi:10.1149/2.0161511jes`.

[38] R. G. M. van der Sman, Simulations of confined suspension flow at multiple length scales, Soft Matter 5 (2009) 4376–4387. `doi:10.1039/B915749M`.

[39] J. Smiatek, F. Schmid, Mesoscopic Simulations of Electroosmotic Flow and Electrophoresis in Nanochannels, Comput. Phys. Commun. 182 (9) (2011) 1941 – 1944, computer Physics Communications Special Edition for Conference on Computational Physics Trondheim, Norway, June 23-26, 2010. `doi:10.1016/j.cpc.2010.11.021`.

[40] R. Wang, J.-S. Wang, G.-R. Liu, J. Han, Y.-Z. Chen, Simulation of DNA electrophoresis in systems of large number

of solvent particles by coarse-grained hybrid molecular dynamics approach, J. Comput. Chem. 30 (4) (2009) 505–513. `doi:10.1002/jcc.21081`.

[41] M. Kuron, G. Rempfer, F. Schornbaum, M. Bauer, C. Godenschwager, C. Holm, J. de Graaf, Moving charged particles in lattice Boltzmann-based electrokinetics, J. Chem. Phys. 145 (21) (2016) 214102. `doi:10.1063/1.4968596`.

[42] J. S. Park, D. Saintillan, Direct Numerical Simulations of Electrophoretic Deposition of Charged Colloidal Suspensions, in: R. C. A. R. Boccaccini, O. Van der Biest, J. Dickerson (Eds.), Key Engineering Materials, Vol. 507, Trans Tech Publ, 2012, pp. 47–51. `doi:10.4028/www.scientific.net/KEM.507.47`.

[43] J.-P. Hsu, C.-H. Chou, C.-C. Kuo, S. Tseng, R. Wu, Electrophoresis of an arbitrarily oriented toroid in an unbounded electrolyte solution, Colloids Surf., B 82 (2) (2011) 505–512. `doi:10.1016/j.colsurfb.2010.10.009`.

[44] J.-P. Hsu, M.-H. Ku, Boundary effect on electrophoresis: finite cylinder in a cylindrical pore, J. Colloid Interface Sci. 283 (2) (2005) 592–600. `doi:10.1016/j.jcis.2004.09.004`.

[45] J.-P. Hsu, L.-H. Yeh, Electrophoresis of Two Identical Rigid Spheres in a Charged Cylindrical Pore, J. Phys. Chem. B 111 (10) (2007) 2579–2586. `doi:10.1021/jp068407z`.

[46] J. H. Masliyah, S. Bhattacharjee, Electrokinetic and Colloid Transport Phenomena, John Wiley & Sons, Inc., 2006.

[47] S. Tseng, C.-H. Huang, J.-P. Hsu, Electrophoresis of two spheres: Influence of double layer and van der Waals interactions, J. Colloid Interface Sci. 451 (2015) 170–176. `doi:10.1016/j.jcis.2015.03.060`.

[48] R. Schmitz, V. Starchenko, B. Dünweg, Computer simulation of electrokinetics in colloidal systems, Eur. Phys. J. Spec. Topics 222 (11) (2013) 2873–2880. `doi:10.1140/epjst/e2013-02063-2`.

[49] M. Baptista, R. Schmitz, B. Dünweg, Simple and robust solver for the Poisson-Boltzmann equation, Phys. Rev. E 80 (2009) 016705. `doi:10.1103/PhysRevE.80.016705`.

[50] H. H. Hu, N. A. Patankar, M. Y. Zhu, Direct Numerical Simulations of Fluid-Solid Systems Using the Arbitrary Lagrangian-Eulerian Technique, J. Comput. Phys. 169 (2) (2001) 427–462. `doi:10.1006/jcph.2000.6592`.

[51] T. J. R. Hughes, W. K. Liu, T. K. Zimmermann, Lagrangian-Eulerian finite element formulation for incompressible viscous flows, Compu. Meth. Appl. Mech. and Engin. 29 (3) (1981) 329–349. `doi:10.1016/0045-7825(81)90049-9`.

[52] C. Ye, D. Li, 3-D transient electrophoretic motion of a spherical particle in a T-shaped rectangular microchannel, J. Colloid Interface Sci. 272 (2) (2004) 480–488. `doi:10.1016/j.jcis.2003.11.014`.

[53] H. Tanaka, T. Araki, Simulation Method of Colloidal Suspensions with Hydrodynamic Interactions: Fluid Particle Dynamics, Phys. Rev. Lett. 85 (2000) 1338–1341. `doi:10.1103/PhysRevLett.85.1338`.

[54] H. Kodama, K. Takeshita, T. Araki, H. Tanaka, Fluid particle dynamics simulation of charged colloidal suspensions, J. Phys. Condens. Matter 16 (10) (2004) L115–L123. `doi:10.1088/0953-8984/16/10/L01`.

[55] T. Araki, H. Tanaka, Physical principle for optimizing electrophoretic separation of charged particles, Europhys. Lett. 82 (1) (2008) 18004. `doi:10.1209/0295-5075/82/18004`.

[56] Y. Nakayama, R. Yamamoto, Simulation method to resolve hydrodynamic interactions in colloidal dispersions, Phys. Rev. E 71 (2005) 036707. `doi:10.1103/PhysRevE.71.036707`.

[57] K. Kim, Y. Nakayama, R. Yamamoto, Direct Numerical Simulations of Electrophoresis of Charged Colloids, Phys. Rev. Lett. 96 (2006) 208302. `doi:10.1103/PhysRevLett.96.208302`.

[58] C. Shih, R. Yamamoto, Dynamic electrophoresis of charged colloids in an oscillating electric field, Phys. Rev. E 89 (2014) 062317. `doi:10.1103/PhysRevE.89.062317`.

[59] X. Luo, A. Beskok, G. E. Karniadakis, Modeling Electrokinetic Flows by the Smoothed Profile Method, J. Comput. Phys. 229 (10) (2010) 3828 – 3847. `doi:10.1016/j.jcp.2010.01.030`.

[60] C. S. Peskin, Flow patterns around heart valves: A numerical method, J. Comput. Phys. 10 (2) (1972) 252–271. `doi:10.1016/0021-9991(72)90065-4`.

[61] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261. `doi:10.1146/annurev.fluid.37.061903.175743`.

[62] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005) 448–476. `doi:10.1016/j.jcp.2005.03.017`.

[63] S. Kang, Direct simulations on the electrophoretic motion of multiple charged particles using an immersed boundary method, Comput. Fluids 73 (2013) 10–23. `doi:10.1016/j.compfluid.2012.12.005`.

[64] V. Lobaskin, B. Dünweg, C. Holm, Electrophoretic mobility of a charged colloidal particle: a computer simulation study, J. Phys. Condens. Matter 16 (38) (2004) S4063–S4073. `doi:10.1088/0953-8984/16/38/021`.

[65] T. Y. Molotilin, V. Lobaskin, O. I. Vinogradova, Electrophoresis of Janus particles: A molecular dynamics simulation study, J. Chem. Phys. 145 (24) (2016) 244704. `doi:10.1063/1.4972522`.

[66] A. Chatterji, J. Horbach, The role of effective charges in the electrophoresis of highly charged colloids, J. Phys. Condens. Matter 22 (49) (2010) 494102. `doi:10.1088/0953-8984/22/49/494102`.

[67] O. A. Hickey, C. Holm, J. Smiatek, Lattice-Boltzmann simulations of the electrophoretic stretching of polyelectrolytes: The importance of hydrodynamic interactions, J. Chem. Phys. 140 (16) (2014) 164904. `doi:10.1063/1.4872366`.

[68] F. Capuani, I. Pagonabarraga, D. Frenkel, Discrete solution of the electrokinetic equations, J. Chem. Phys. 121 (2) (2004) 973–986. `doi:10.1063/1.1760739`.

[69] G. Giupponi, I. Pagonabarraga, Determination of the zeta potential for highly charged colloidal suspensions, Phil. Trans. R. Soc. A: Mathematical, Physical and Engineering Sciences 369 (2011) 2546–2554. `doi:10.1098/rsta.2011.0024`.

[70] G. K. Batchelor, An Introduction to Fluid Dynamics, Cambridge Univ. Press, 1979.

[71] M. A. van der Hoef, M. Ye, M. van Sint Annaland, A. T. Andrews, S. Sundaresan, J. A. M. Kuipers, Multi-Scale Modeling of Gas-Fluidized Beds, in: G. B. Marin (Ed.), Computational Fluid Dynamics, Vol. 31 of Advances in Chemical Engineering, Elsevier, 2006, pp. 65–149. `doi:10.1016/S0065-2377(06)31002-2`.

[72] L. D. Landau, E. M. Lifshitz, Fluid Mechanics, Second Edition: Volume 6 (Course of Theoretical Physics), 2nd Edition,

Vol. 6 of Course of theoretical physics / by L. D. Landau and E. M. Lifshitz, Vol. 6, Butterworth-Heinemann, 1987.

[73] P. J. Dellar, Bulk and shear viscosities in lattice Boltzmann equations, Phys. Rev. E 64 (3) (2001) 031203. `doi:10.1103/PhysRevE.64.031203`.

[74] J. Happel, H. Brenner, Low Reynolds number hydrodynamics: with special applications to particulate media, Vol. 1, Springer, 1983.

[75] D. Bartuschat, Direct Numerical Simulation of Particle-Laden Electrokinetic Flows on High Performance Computers, Ph.D. thesis, University of Erlangen-Nürnberg (2016).
URL `https://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/7298`

[76] G. G. Stokes, On the effect of the internal friction of fluids on the motion of pendulums, Vol. 9, Pitt Press, 1851.

[77] J. Happel, E. Bart, The settling of a sphere along the axis of a long square duct at low Reynolds' number, Appl. Sci. Res. 29 (1) (1974) 241–258. `doi:10.1007/BF00384149`.

[78] A. Miyamura, S. Iwasaki, T. Ishii, Experimental wall correction factors of single solid spheres in triangular and square cylinders, and parallel plates, Int. J. Multiph. Flow 7 (1) (1981) 41–46. `doi:10.1016/0301-9322(81)90013-6`.

[79] R. W. O'Brien, L. R. White, Electrophoretic mobility of a spherical colloidal particle, J. Chem. Soc., Faraday Trans. 2 74 (1978) 1607–1626. `doi:10.1039/F29787401607`.

[80] R. J. Hunter, Zeta Potential in Colloid Science: Principles and Applications, Academic Press, London, 1981.

[81] P. Debye, E. Hückel, Zur Theorie der Elektrolyte. I. Gefrierpunktserniedrigung und verwandte Erscheinungen, Phys. Z 24 (9) (1923) 185–206.

[82] D. C. Henry, The cataphoresis of suspended particles. Part I. The equation of cataphoresis, Proc. R. Soc. Lond. A 133 (821) (1931) 106–129. `doi:10.1098/rspa.1931.0133`.

[83] H. Ohshima, A Simple Expression for Henry's Function for the Retardation Effect in Electrophoresis of Spherical Colloidal Particles, J. Colloid Interface Sci. 168 (1) (1994) 269–271. `doi:10.1006/jcis.1994.1419`.

[84] H. Ohshima (Ed.), Theory of Colloid and Interfacial Electric Phenomena, Vol. 12 of Interface Science and Technology, Elsevier, 2006. `doi:10.1016/S1573-4285(06)80022-0`.

[85] D. J. Shaw, Introduction to Colloid and Surface Chemistry, 4th Edition, Butterworth-Heinemann, 1992.

[86] H. Ohshima, T. W. Healy, L. R. White, Accurate analytic expressions for the surface charge density/surface potential relationship and double-layer potential distribution for a spherical colloidal particle, J. Colloid Interface Sci. 90 (1) (1982) 17–26. `doi:10.1016/0021-9797(82)90393-9`.

[87] A. L. Loeb, J. T. G. Overbeek, P. H. Wiersema, The Electrical Double Layer Around a Spherical Colloid Particle, J. Electrochem. Soc. 108 (12) (1961) 269C. `doi:10.1149/1.2427992`.

[88] D. Hänel, Molekulare Gasdynamik, Springer, 2004.

[89] S. Chapman, T. G. Cowling, The Mathematical Theory of Non-Uniform Gases, 3rd Edition, Cambridge Univ. Press, 1990.

[90] D. A. Wolf-Gladrow, Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction, no. 1725 in Lattice-gas Cellular Automata and Lattice Boltzmann Models: An Introduction, Springer, 2000.

[91] D. d'Humières, Generalized lattice-Boltzmann equations, in: Rarefied Gas Dynamics: Theory and Simulations, Vol. 159 of Prog. Astronaut. Aeronaut., AIAA, 1992, pp. 450–458. `doi:10.2514/5.9781600866319.0450.0458`.

[92] X. He, L.-S. Luo, Lattice Boltzmann Model for the Incompressible Navier-Stokes Equation, J. Stat. Phys. 88 (3) (1997) 927–944. `doi:10.1023/B:JOSS.0000015179.12689.e4`.

[93] Y. H. Qian, D. d'Humières, P. Lallemand, Lattice BGK Models for Navier-Stokes Equation, Europhys. Lett. 17 (6) (1992) 479. `doi:10.1209/0295-5075/17/6/001`.

[94] J. D. Sterling, S. Chen, Stability Analysis of Lattice Boltzmann Methods, J. Comput. Phys. 123 (1) (1996) 196–206. `doi:10.1006/jcph.1996.0016`.

[95] 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 (5) (2011) 056710. `doi:10.1103/PhysRevE.83.056710`.

[96] P. L. Bhatnagar, E. P. Gross, M. Krook, A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems, Phys. Rev. 94 (3) (1954) 511–525. `doi:10.1103/PhysRev.94.511`.

[97] M. Geier, M. Schönherr, A. Pasquali, M. Krafczyk, The cumulant lattice Boltzmann equation in three dimensions: Theory and validation, Comput. Math. Appl. 70 (4) (2015) 507–547. `doi:10.1016/j.camwa.2015.05.001`.

[98] F. J. Higuera, S. Succi, R. Benzi, Lattice Gas Dynamics with Enhanced Collisions, Europhys. Lett. 9 (4) (1989) 345. `doi:10.1209/0295-5075/9/4/008`.

[99] I. Ginzburg, F. Verhaeghe, D. d'Humières, Study of Simple Hydrodynamic Solutions with the Two-Relaxation-Times Lattice Boltzmann Scheme, Commun. Comput. Phys. 3 (3) (2008) 519–581.

[100] L.-S. Luo, Analytic Solutions of Linearized Lattice Boltzmann Equation for Simple Flows, J. Stat. Phys. 88 (3/4) (1997) 913–926. `doi:10.1023/B:JOSS.0000015178.19008.78`.

[101] A. J. C. Ladd, R. Verberg, Lattice-Boltzmann Simulations of Particle-Fluid Suspensions, J. Stat. Phys. 104 (5) (2001) 1191–1251. `doi:10.1023/A:1010414013942`.

[102] Z. Guo, C. Zheng, B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, Phys. Rev. E 65 (4) (2002) 046308. `doi:10.1103/PhysRevE.65.046308`.

[103] C. Rettinger, U. Rüde, A comparative study of fluid-particle coupling methods for fully resolved lattice Boltzmann simulations, Comput. Fluids 154 (2017) 74–89. `doi:10.1016/j.compfluid.2017.05.033`.

[104] E. Fattahi, C. Waluga, B. Wohlmuth, U. Rüde, M. Manhart, R. Helmig, Lattice Boltzmann methods in porous media simulations: From laminar to turbulent flow, Comput. Fluids 140 (2016) 247 – 259. `doi:10.1016/j.compfluid.2016.10.007`.

[105] S. Bogner, S. Mohanty, U. Rüde, Drag correlation for dilute and moderately dense fluid-particle systems using the lattice

Boltzmann method, Int. J. Multiph. Flow 68 (2015) 71–79. `doi:10.1016/j.ijmultiphaseflow.2014.10.001`.

[106] P. Wesseling, Introduction To Multigrid Methods, R. T. Edwards, 2004.

[107] P. Knabner, L. Angermann, Numerical Methods for Elliptic and Parabolic Partial Differential Equations, Texts in Applied Mathematics, Springer, 2003.

[108] B. I. des Poids et Mesures, The International System of Units (SI), 8th Edition, BIPM, 2006 [cited May 2018].
URL `http://www.bipm.org/en/publications/si-brochure/`

[109] I. 80000-1:2009, Quantities and units – Part 1: General, ISO/IEC, 2009 [cited May 2018].
URL `https://www.iso.org/obp/ui/#iso:std:30669:en`

[110] S. Donath, Wetting Models for a Parallel High-Performance Free Surface Lattice Boltzmann Method, Ph.D. thesis, University of Erlangen-Nürnberg (2011).
URL `http://www.dr.hut-verlag.de/978-3-8439-0066-9.html`

[111] D. Bartuschat, D. Ritter, U. Rüde, Parallel multigrid for electrokinetic simulation in particle-fluid flows, in: High Performance Computing and Simulation (HPCS). Madrid 2012, IEEE, 2012, pp. 374–380. `doi:10.1109/HPCSim.2012.6266940`.

[112] R. Ramadugu, S. P. Thampi, R. Adhikari, S. Succi, S. Ansumali, Lattice differential operators for computational physics, Europhys. Lett. 101 (5) (2013) 50006. `doi:10.1209/0295-5075/101/50006`.

[113] K. Iglberger, Software design of a massively parallel rigid body framework, Ph.D. thesis, University of Erlangen-Nürnberg (2010).
URL `http://www.dr.hut-verlag.de/978-3-86853-736-9.html`

[114] G. Hager, G. Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press, 2010. `doi:10.1201/EBK1439811924`.

[115] G. Wellein, T. Zeiser, G. Hager, S. Donath, On the single processor performance of simple lattice Boltzmann kernels, Comput. Fluids 35 (8) (2006) 910–919. `doi:10.1016/j.compfluid.2005.02.008`.

[116] L. Hufnagel, Transient simulation of electric double layers in electrokinetic flows with a coupled lattice Boltzmann Link-Flux method, Bachelor thesis, Lehrstuhl für Informatik 10 (Systemsimulation), Universität Erlangen-Nürnberg (Nov. 2014).