University of Stuttgart
Germany

Master's-Thesis

Particle coupling in continuum electrokinetics simulations

Robert Kaufmann

October 22, 2018

Examiner: Prof. Dr. Christian Holm
Institute for Computational Physics
University of Stuttgart
Co-Examiner: Prof. Dr. Siegfried Dietrich
Max-Planck-Institut for Intelligent Systems

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe und ohne unerlaubte Hilfsmittel angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, die den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe, dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen gegenstand eines anderen Prüfungsverfahrens gewesen ist, dass ich sie weder vollständig noch in Teilen bereits veröffentlich habe und, dass der Inhalt des elektronischen Exemplars mit dem des Druckexemplars übereinstimmt.

Stuttgart, 22. Oktober 2018

Robert Kaufmann

Zusammenfassung

Computer Simulationen sind ein wichtiges Werkzeug für Systeme, in denen es an analytischen Lösungen fehlt. Dank der immer weiter steigenden Rechenleistung und der Optimierung einer immer weiter wachsenden Zahl von Simulationsmethoden können wir immer komplexere and größere Systeme simulieren. Ein Themenfeld in dem Simulationen oft genutzt werden ist das der Weichen Materie. Weiche Materie Systeme werden von Interaktionen beherrscht, die sich im Bereich der Thermischen Energie befinden, das heißt Entropie ist ein wichtiger Teil dieser Systeme. Synthetische Materialien, wie Plastik, aber auch organische Substanzen, wie **D**esoxyribo**n**ukleins**ä**ure (DNS), **R**ibo**n**ukleins**ä**ure (RNA) oder Proteine, sind ebenfalls Polymere, die in das Gebiet der Weichen Materie fallen. Gele, Gummi, Flüssigkristalle und kolloidale Suspensionen gehören auch zum Feld der Weichen Materie. Das Verständnis um die räumliche Struktur und Dynamik von Proteinen trägt maßgeblich zum Verstehen ihrer Funktion bei [14]. Das Falten von Proteinen ist eine wichtige Mechanik in jedem Organismus. Die Ansammlung von missgefalteten Proteinen ist Bestandteil von verschiedensten Krankheiten [8]. Biomoleküle wie Proteine oder DNS sind daher ein wichtiger Forschungsgegenstand.
Die Fähigkeit das Verhalten von DNS zu simulieren kann helfen neue Methoden zu finden um zum Beispiel DNS zu sequenzieren. Das Falten von Proteinen besonders für Methoden, bei denen mikrofabrizierte Geräte genutzt werden, spielen Simulation eine wichtige Rolle [20]. Eine aktuell vielversprechende Methode ist die Sequenzierung von DNS mithilfe von Nanoporen, hier wird die DNS durch eine Nanopore transportiert, was den Ionenstrom durch die Pore beeinflusst. Kann man diesen Ionenstrom in die Sequenz der DNS übersetzen, ergibt sich die Möglichkeit schnell und einfach DNS zu sequenzieren. Aksimentiev [1] gibt eine gute Übersicht über dieses Thema. Solche Systeme wurden mit Moleküldynamik-Simulationen untersucht [2] [18], aber es wurden auch Modelle, in denen Teile des Systems, wie das Lösemittel, nur implizit beschrieben sind, genutzt [30].
Für Weiche Materie Systeme müssen Hydrodynamik, Diffusion und Elektrostatische Kräfte für eine große Menge von Atomen korrekt gelöst werden, was dazu führt, dass wir trotz des technologischen Fortschritts nur eher kleine Biomoleküle simulieren können. Modelle, in denen das Lösungsmittel, zum Beispiel Wasser, nur implizit beschreiben wird, benötigen

deutlich weniger Rechenleistung und können es ermöglichen größere Systeme für längere Zeiträume zu simulieren. Die *Lattice Boltzmann* Methode ist eine oft benutzte Kontinuum Methode, die für Weiche Materie Systeme die *Navier-Stokes* Gleichungen reproduziert und damit die Hydrodynamik auf billige Weise implementiert. Ein großes Problem stellen die unterschiedlichen Längen- und Zeitskalen da, die in Systemen mit Partikeln unterschiedlichster Größe, wie zum Beispiel Ionen und Monomere in einer Polyelektrolyte Simulation, auftreten. Eine Möglichkeit dieses Problem zu lösen ist die Ionen auch mit einer Kontinuumsmethode zu behandeln.
Diese Arbeit baut auf eine Methode auf, die zuerst von Capuani et. al. [4] vorgeschlagen wurde und von Rempfer [24] in ESPResSO implementiert wurde. Mit dieser Implementierung wurden bereits erfolgreich Systeme wie die Elektrophorese einer geladenen Kugel simuliert [17]. ESPResSO ist ein Software Paket, das hauptsächlich am Institut für Computerphysik der Universität Stuttgart von der Forschungsgruppe von Prof. Dr. Christian Holm entwickelt wird [3] [19].
In den folgenden Kapiteln wird zunächst das von Rempfer implementierte Modell erklärt und anschließend um die Thermalisierung der Ionen Ströme erweitert, die bisher nicht enthalten war, aber in Weiche Materie Systemen eine Rolle spielt. In Kapitel 2.4 leiten wir den Term für die thermischen Fluktuationen, ausgehend von der unkorrelierten Brownschen Bewegung der individuellen Teilchen, her und stellen eine Gleichung für die Evolution der Dichte auf. Wir implementieren eine neue Diskretisierung in Kapitel 3 und validieren diese und den neuen Fluktuationsterm in Kapitel 5. In Kapitel 6 wiederholen wir Polymer Simulation von Schoell [27], um zu überprüfen, ob die Thermalisierung der Ionen die Ergebnisse beeinflusst. Danach werden Probleme des Modells und deren Auswirkung auf die Polymer Simulationen diskutiert. Wir beenden die Arbeit mit einer kurzen Zusammenfassung der Ergebnisse und einem Ausblick für zukünftige Arbeiten.

Contents

Chapter 1

Introduction

Computer simulations are an important tool for any system or field which lacks analytical solutions. With the exponential increase in computational power and an ever increasing amount of simulation methods we are able to model and simulate bigger and more complicated systems. A popular field in which simulations are utilized is soft matter. Soft matter systems are governed by interactions in the range of the thermal energy, which means entropy plays an important part to properly simulate them. Synthetic materials, like plastics, but also any organic substances, such as **D**eoxyribo**N**ucleic **A**cid, **R**ibo**N**ucleic **A**cid, or proteins consist of polymers which fall into this research field. Gels, rubbers, liquid crystals, colloidal suspensions, all fall into the field of soft matter. Understanding for example the conformational dynamics of proteins is important to explain how they function [14]. The folding and unfolding of proteins is a very important mechanic in any living system and aggregations of misfolded proteins are a common feature in several debilitating diseases [8]. The importance of these biomolecules makes them a crucial research topic.
Being able to simulate the behaviour and dynamics of DNA can help with finding new or improving methods to separate or sequence DNA. Especially for the development of methods making use of micro-fabricated devices computational physics have played an important part [20]. A promising method is using nanopores to sequence DNA, here the modulations in ionic current caused by DNA moving through a nanopore are analysed. If we were able to translate these currents into the sequence of the DNA, we would be able to easily read out the DNA of any organism. An overview over this topic can be found in Aksimentiev [1]. This type of system has been investigated with atomistic and coarse-grained **M**olecular **D**ynamics simulations [2] [18] but also using models with implicit solvents [30]. Complex systems like these require correctly modeling diffusion, hydrodynamic and electrostatic interactions for a large amount of atoms and while technology has improved we still can only simulate very small biomolecules on the atomic level. Modeling the solvent of such a system implicitly is a potent way to reduce computational cost, which is

limiting all-atom simulations. A particular successful approach in this context is the *Lattice Boltzmann* method, which is a lattice based continuum method that reproduces the Navier-Stokes equations in limits usually valid for soft matter systems. While this allows the inclusion of hydrodynamics in computational cheap way, it does not solve the problem of very different time and length scales in systems that feature differently sized particles such as ions and coarse-grained monomers in a polyelectrolyte simulation. Treating the ions using a continuum description as well can solve this problem.
This thesis builds on a method first proposed by Capuani, Pagonabarraga and Frenkel in 2004 [4] which was later implemented by Rempfer [24] into the **E**xtensible **S**imulation **P**ackage for **R**esearch on **S**oft Matter (ESPResSo). This electrokinetic solver was already successfully applied to different systems such as the Electrophoresis of a charged sphere [17]. ESPResSO is a software package mainly developed by Prof. Dr. Christian Holm's resarch group at the institute for Computational Physics at the University in Stuttgart [3] [19].
In the following chapters we give a brief overview over the model implemented by Rempfer and proceed by adding the thermalisation of the ion fluxes, which was so far missing from the model but can play an important role in soft matter systems. In Chap. 2.4 we derivate the noise term starting from uncorrelated Brownian motion for the individual particles and establish an equation for the evolution of the density following [5]. We also implement a new discretization stencil in Chap. 3 and after verifying the behaviour of the newly added fluctuations in Chap. 5 we revisit polymer simulations done by Schoell [27] in hopes of improving the results. Afterwards we will analyse existing problems with the model and close with a short summary and outlook in the last chapter.

Chapter 2

The Electrokinetic Model

2.1 The Electrokinetic Equations

This thesis builds upon the *Electrokinetics* feature of ESPresSo implemented by Rempfer [24]. The systems we treat with this feature consist of one or more diffusive species and a fluid, a simple example would be any salt dissolved in water. Through electrostatic and hydrodynamic interactions the diffusive species and the fluid, can be coupled to explicitly simulated MD particles, thus allowing us to simulate complex systems such as DNA dissolved in a fluid with added salt.
The following explanation of the theory behind it is mainly taken from [24]. We consider a system of one or more diffusive species in a fluid. In this continuum description we define a scalar number density field $p_k(\boldsymbol{r}, t)$ and density flux $j_k(\boldsymbol{r}, t)$ for every species k. The flux consists of terms caused by diffusion and by advection

$$\boldsymbol{j}_k = \boldsymbol{j}_k^{\text{adv.}} + \boldsymbol{j}_k^{\text{diff.}}. \tag{2.1}$$

The time evolution of the density is governed by the continuity equation

$$\frac{\partial n_k}{\partial t} = -\nabla \cdot \boldsymbol{j}_k, \tag{2.2}$$

which ensures mass conservation if satisfied locally. In order to derive the diffusive flux $\boldsymbol{j}_k^{\text{diff.}}$ we treat the species as an ideal gas with added electrostatic interactions. In this description we can obtain the local free energy and define the diffusive flux in terms of a thermodynamic potential. We start by defining the local free energy f of the system as

$$f(c_k(\boldsymbol{r}) = \sum_k \underbrace{k_\text{b} T c_k(\boldsymbol{r})[\log\left(\Lambda_k^3 c_k(\boldsymbol{r})\right) - 1]}_{\text{ideal gas contribution}} + \underbrace{z_k c_k(\boldsymbol{r}) \Phi(\boldsymbol{r})}_{\text{electrostatic}} \tag{2.3}$$

with the *Boltzmann constant* k_b, temperature T, de Broglie wavelength Λ, charge of a particle q_k and the electrostatic potential Φ.

The diffusive flux can be seen as a drift towards local equilibrium and can therefore be expressed through a thermodynamic driving force. The chemical potential $\mu_k(\boldsymbol{r})$ is the associated thermodynamic potential. The diffusive flux can be written as

$$j_k^{\text{diff.}} = -\nu_k \cdot n_k \nabla \mu_k, \tag{2.4}$$

where ν_k is the mobility. Together with

$$\mu_k(\boldsymbol{r}) = \frac{\partial f(n_k)}{\partial n_k} \tag{2.5}$$

the diffusive flux becomes

$$\begin{aligned} j_k^{\text{diff.}} &= -k_{\text{B}} T \nu_k \nabla n_k - \nu_k q_k n_k \nabla \Phi \\ &= -D_k \nabla n_k - \nu_k q_k n_k \nabla \Phi, \end{aligned} \tag{2.6}$$

where $D_k = \nu_k k_{\text{B}} T$ is the diffusion coefficient fulfilling the *Einstein-Smoluchowski* relation. Now we consider the fluid and the advective flux that is caused by the moving fluid transporting the species. Under the assumption that particles of a species instantaneously assume the advective velocity $\boldsymbol{v}$ the related flux is

$$j_k^{\text{adv.}} = n_k \boldsymbol{v}. \tag{2.7}$$

Like in the overdamped limit of a langevin simulation inertial effects of single particles are ignored.

Putting together equations 2.2, 2.6 and 2.7 yields the following term for the time evolution of the densities

$$\partial_t n_k = -\nabla (j_k^{\text{diff.}} + j_k^{\text{adv.}}) = \nabla (D_k \nabla n_k + \nu_k q_k n_k \nabla \Phi - n_k \boldsymbol{v}). \tag{2.8}$$

The electrostatic potential is derived under the assumption that magnetic and electrodynamic effects can be neglected. This is justified for soft matter systems like electrolytic solutions which are characterised by slow dynamics and low current densities. In such a system the electric field is given by Poissons's equation

$$\nabla \cdot (\epsilon \boldsymbol{E}) = \sum_k q_k n_k \tag{2.9}$$

with ϵ being the permittivity of the medium. In this work we only consider constant permittivity. The electrostatic field can be written as $\boldsymbol{E} = -\nabla \Phi$ and the potential Φ is defined through Poisson's equation

$$\Delta \Phi = \frac{1}{\epsilon} \sum_k q_k n_k = -4\pi l_{\text{B}} k_{\text{B}} T \sum_k q_k n_k \tag{2.10}$$

where $l_\mathrm{B} = \frac{e^2}{4\pi\epsilon k_\mathrm{B} T}$ is the Bjerrum length. To complete the model we need a description for the advective velocity $\boldsymbol{v}$. We start with the *Navier-Stokes* equations for the fluid. They can be derived by inserting the stress tensor of an incompressible Newtonian fluid into the conservation of momentum

$$\nabla \cdot \mathbf{\Pi} = -\frac{\partial}{\partial t}(\rho \boldsymbol{u}), \tag{2.11}$$

with the stress tensor $\mathbf{\Pi}$, the local density$p$ and the fluid velocity $\boldsymbol{u}$. Assuming an incompressible Newtonian fluid the *Navier-Stokes*-equations are

$$\begin{aligned} \rho\frac{\partial}{\partial t}\boldsymbol{u} + \rho(\boldsymbol{u}\nabla)\boldsymbol{u} &= -\nabla p + \eta\Delta\boldsymbol{u} + \boldsymbol{f} \\ \rho\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{u}) &= 0 \end{aligned} \tag{2.12}$$

with hydrodynamic pressure p, dynamic viscosity η and external force density $\boldsymbol{f}$. The behaviour of the flow is indicated by the dimensionless *Reynold's* number $\mathrm{Re} = \frac{\rho\bar{u}l}{\eta}$, where $\bar{u}$ is the average fluid velocity and l a characteristic length scale of the system. For low *Reynold's* numbers the viscous forces are dominant, in this limit and in a stationary reference frame Eq. 2.12 becomes the *Stokes* equation

$$\begin{aligned} \eta\Delta\boldsymbol{u} &= \nabla p - \boldsymbol{f}. \\ \nabla\cdot\boldsymbol{v} &= 0 \end{aligned} \tag{2.13}$$

where η is the sheer viscosity. Detailed in Rempfer et. al. [25] we extend the commonly used hydrodynamic driving force $\boldsymbol{f} = -\sum_k q_k e \rho_k \nabla\Phi$ by a term corresponding to the ideal gas pressure of the ion species yielding

$$\boldsymbol{f} = -\sum_k (k_\mathbf{B} T \nabla\rho_k + q_k e \rho_k \nabla\Phi). \tag{2.14}$$

This choice reduces problems with spurious flow because the force density acting on the fluid now vanishes in equilibrium. With this force density the *Stokes* equations become

$$\begin{aligned} \eta\Delta\boldsymbol{v} &= \nabla p - \sum_k (k_\mathbf{B} T \nabla\rho_k + q_k e \rho_k \nabla\Phi). \\ \nabla\cdot\boldsymbol{v} &= 0 \end{aligned} \tag{2.15}$$

2.2 Lattice-Boltzmann

The fluid is handled with a lattice based continuum model. Specifically the *Lattice-Boltzmann* (LB) method [16] is used to solve the *Stokes* equation. The model introduces a regular cubic grid with populations of discrete velocities sitting on the nodes. Each velocity is chosen such that it transports a particle from one node to a neighbouring node during

one time step. The LB method can be divided in two steps, the streaming step and the collision step. During the streaming step populations of the discrete velocities are transferred to the associated neighbours and during the collision step the populations relax towards an equilibrium distribution. Figure 2.1 shows a visualization of the two steps for a 2D grid.

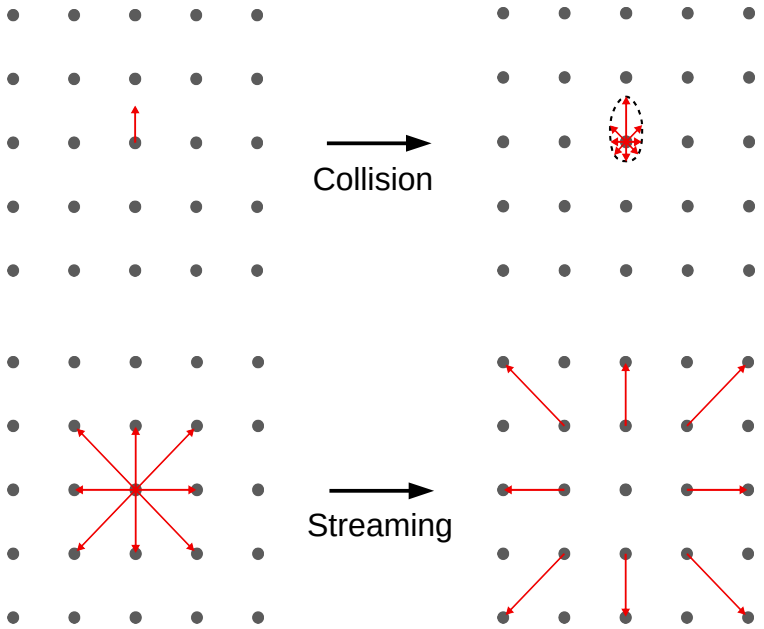

Figure 2.1: Visualization of the streaming and collision step of the LB method in 2D.

The LB method can be obtained from a discretisation of the *Boltzmann* equation, which describes the evolution of the phase space distribution of the particles. We use the it to numerically solve equation 2.15. The phase space distribution function $f(\boldsymbol{x}, \boldsymbol{p}, t)$ denotes the number density of particles at position $\boldsymbol{x}$ with momentum $\boldsymbol{p}$ at time t and its time evolution is described by

$$\frac{d}{dt} f(\boldsymbol{r}, \boldsymbol{v}, t) = \left(\boldsymbol{v} \cdot \nabla_{\boldsymbol{r}} + \frac{1}{m} \boldsymbol{F} \cdot \nabla_{\boldsymbol{v}} + \frac{\partial}{\partial t} \right) f(\boldsymbol{r}, \boldsymbol{v}, t) = \frac{1}{\tau} (f^{\mathrm{eq}}(\boldsymbol{v} - f(\boldsymbol{r}, \boldsymbol{v}, t)). \tag{2.16}$$

where m is the mass of a single particle, $\boldsymbol{F}$ is an externally applied force, τ the relaxation time and f^{eq} the equilibrium distribution which is known to be a Maxwell-Boltzmann distribution. Discretizing $\frac{d}{dt} f(\boldsymbol{r}, \boldsymbol{v}, t)$ using a finite number of velocities and a regular cubic grid leads to

$$f_i(\boldsymbol{r} + \boldsymbol{c}_i \Delta t, t + \Delta t) = f_i(\boldsymbol{r}, t) + \sum_k L_{ik} f_k^{\mathrm{eq}} - f_k(\boldsymbol{r}, t) + \Psi_i, \tag{2.17}$$

where Ψ_i is the change in population caused by external forces and the relaxation τ is replaced with the Multi-Relaxation Time collision operator L. Using this linear operator,

which can be expressed as a matrix, we can set the fluid's bulk and sheer viscosity individually. In the continuum limit with the right choice of the collision operator the LB method reproduces the desired *Stokes* equation. The *Electrokinetics* feature makes use of the in ESPResSo implemented D3Q19 LB scheme, which uses 19 discrete velocities on a 3D grid (see Fig. 2.2). For a more detailed explanation of the LB method and the collision operator used see Dünweg [10].

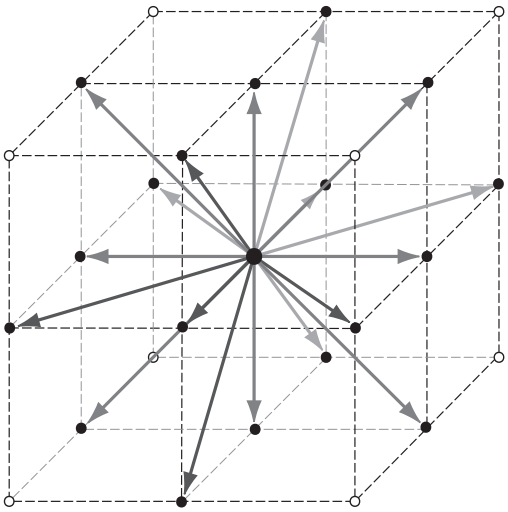

Figure 2.2: A node of a D3Q19 LB grid, showing the 19 discrete velocities. Image taken from rempfer [24]

2.3 Electrostatic interaction

The electrostatic interaction is handled through an electrostatic potential Φ. Poisson's equation gives us

$$\nabla^2\Phi(\boldsymbol{r}) = -4\pi l_\mathrm{B} k_\mathrm{B} T \sum_k q_k n_k(\boldsymbol{r}). \tag{2.18}$$

This is a single linear, elliptic, Partial Differential Equation in three dimensions. Since the Lattice-Boltzmann method is discretized on a regular cubic grid it lends itself to also discretize the electrostatic interaction on this grid. We use a finite difference scheme, where the derivative is approximated as the finite difference between points with spacing a. Only considering the nearest neighbours yields

$$-4\pi l_\mathrm{B} k_\mathrm{B} T \sum_k q_k n_k(\boldsymbol{r}) = \sum_{\boldsymbol{c}_i} (\Phi(\boldsymbol{r}+\boldsymbol{c}_i) - \Phi(\boldsymbol{r})), \tag{2.19}$$

where $\boldsymbol{r}+\boldsymbol{c}_i$ is the location of the neighbours. For our regular grid in three dimensions there are 6 nearest neighbours and node $\boldsymbol{r}$ itself, this is a 7 point stencil. In order to solve

this system of linear equations we make use of Fourier transformations since the equations will decouple in Fourier space. The following steps are used to solve the system [24]:

1. Using an FFT algorithm to calculate the Fourier Space representation $\hat{\rho}(\boldsymbol{k})$ of the charge density $\rho(\boldsymbol{r})$.

2. A shift in real space is a multiplication with a scalar in Fourier space, shifting $f(x, y, z)$ by (a, b, c) for example yields

$$\begin{aligned} & f(x+a, y+b, z+c) \\ & \quad = \frac{1}{\sqrt{N_x N_y N_z}} \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} \sum_{k_z=0}^{N_z-1} e^{2\pi i\left(\frac{k_x(x+a)}{N_x}+\frac{k_y(y+b)}{N_y}+\frac{k_z(z+c)}{N_z}\right)} \hat{f}(\boldsymbol{k}) \\ & \quad = \frac{1}{\sqrt{N_x N_y N_z}} \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} \sum_{k_z=0}^{N_z-1} e^{2\pi i\left(\frac{k_x x}{N_x}+\frac{k_y y}{N_y}+\frac{k_z z}{N_z}\right)} \left[e^{2\pi i\left(\frac{k_x a}{N_x}+\frac{k_y b}{N_y}+\frac{k_z c}{N_z}\right)} \hat{f}(\boldsymbol{k}) \right] . \end{aligned} \tag{2.20}$$

   Utilizing this theorem the finite difference approximation and the 7-point stencil allow us to write

$$\Phi(\boldsymbol{k}) = \frac{-2\pi h^2 l_{\mathrm{B}} k_{\mathrm{B}} T}{\cos(\frac{2\pi k_x}{N_x}) + \cos(\frac{2\pi k_x}{N_x}) + \cos(\frac{2\pi k_x}{N_x}) - 3} \hat{\rho}(\boldsymbol{k}), \tag{2.21}$$

   where we switched to Cartesian coordinates and $N_{x,y,z}$ is the number of lattice nodes in dimension x, y, z.

3. Applying an FFT back transform to calculate the electrostatic potential in real space $\Phi(\boldsymbol{r})$.

2.4 Noise Term

The model described in the previous sections is not fully thermalised yet. In soft matter systems such as the later investigated polyelectrolytes the interactions at play are in the range of the thermal energy, which means thermal fluctuations should be considered. We introduce a noise term in the density flux to solve this problem. We start by considering a system of N identical particles all undergoing independent Brownian motion. We also include a pair potential $V(r)$ and an external one-body potential $\Phi(\boldsymbol{r_i}, t)$. The particles obey

$$\frac{\mathrm{d}\boldsymbol{r_i}}{\mathrm{d}t} = \eta_i(t) - \nabla_i \boldsymbol{\Phi}(\boldsymbol{r_i}, t) - \sum_{j=1}^{N} \nabla \boldsymbol{V}(|\boldsymbol{r_i} - \boldsymbol{r_j}|), \tag{2.22}$$

where $\eta_i(t)$ is a uncorrelated Gaussian random force defined through

$$\langle \eta_i^{\mu}(t) \eta_j^{\nu}(t') \rangle = 2D\delta_{ij}\delta^{\mu\nu}\delta(t-t'), \quad \langle \boldsymbol{\eta_i}(t) \rangle = 0. \tag{2.23}$$

Following [5] we will derive a equation for the global density operator

$$n(\boldsymbol{r},t)=\sum_{i=1}^{N}n_i(\boldsymbol{r},t)=\sum_{i=1}^{N}\delta(\boldsymbol{r_i}(t)-\boldsymbol{r}) \tag{2.24}$$

with $n_i(\boldsymbol{r},t)=\delta(\boldsymbol{r_i}(t)-\boldsymbol{r})$ being the density function of a single particle. With f being an arbitrary function the definition of density allows us to write

$$f(\boldsymbol{r}_i(t))=\int \mathrm{d}^3r\, n_i(\boldsymbol{r},t)f(\boldsymbol{r}) \tag{2.25}$$

and

$$\frac{\mathrm{d}f(\boldsymbol{r_i})}{\mathrm{d}t}=\int \mathrm{d}^3r\,\frac{\partial n_i(\boldsymbol{r},t)}{\partial t}f(\boldsymbol{r}). \tag{2.26}$$

Equation 2.25 can be expanded over the next small time step dt using stochastic Ito calculus [23]. Ito calculus extends methods of calculus to stochastic processes, *Ito's Lemma* can be seen as the stochastic counterpart of the chain rule, in differential form with $f=f(t,X)$, where X denotes Brownian motion, it is written as

$$df(t,X(t))=\frac{df(t,X(t))}{\partial t}dt+\frac{1}{2}\frac{\partial^2 f}{\partial X^2}dt+\frac{\partial f}{\partial X}dX. \tag{2.27}$$

Applying this to Equation 2.25 yields

$$\begin{aligned}\frac{\mathrm{d}f(\boldsymbol{r_i})}{\mathrm{d}t}=\int \mathrm{d}^3r\, f(\boldsymbol{r})\Big(&-\nabla\cdot(n_i(\boldsymbol{r},t)\eta_i(t))+\nabla\cdot(n_i(\boldsymbol{r},t)\nabla\boldsymbol{\Phi}(\boldsymbol{r_i},t))\\&+\nabla n_i(\boldsymbol{r},t)\cdot\left(\sum_{j=1}^{N}\nabla V(|\boldsymbol{r}-\boldsymbol{r}_j|)\right)+D\nabla^2 n_i(\boldsymbol{r},t)\Big).\end{aligned} \tag{2.28}$$

Comparing equation 2.26 to 2.28 and summing over the single particles i we arrive at

$$\begin{aligned}\frac{\partial n(\boldsymbol{r},t)}{\partial t}=&\xi(\boldsymbol{r},t)+\nabla\cdot n(\boldsymbol{r},t)\nabla\boldsymbol{\Phi}(\boldsymbol{r_i},t)\\&+\nabla\cdot\left(p(\boldsymbol{r},t)\int \mathrm{d}^3r'\, n(\boldsymbol{r'},t)\nabla V(|\boldsymbol{r}-\boldsymbol{r'}|)\right)+D\nabla^2 n(\boldsymbol{r},t),\end{aligned} \tag{2.29}$$

where the noise term $\xi(\boldsymbol{r},t)$ is defined as

$$\xi(\boldsymbol{r},t)=-\sum_{i=1}^{N}\nabla\cdot(n_i(\boldsymbol{r},t)\eta_i(t)) \tag{2.30}$$

with correlation function

$$\langle\xi(\boldsymbol{r},t)\xi(\boldsymbol{r'},t')\rangle=2D\delta(t-t')\sum_{i=1}^{N}\nabla_r\cdot\nabla_{r'}(n_i(\boldsymbol{r},t)n_i(\boldsymbol{r'},t)). \tag{2.31}$$

The correlation function can be rewritten using the properties of the Dirac delta function yielding

$$\langle \xi(\boldsymbol{r},t)\xi(\boldsymbol{r'},t')\rangle = 2D\delta(t-t')\nabla_r \cdot \nabla_{r'}(\delta(\boldsymbol{r}-\boldsymbol{r'})n_i(\boldsymbol{r},t)). \tag{2.32}$$

We introduce a global noise field

$$\xi'(\boldsymbol{r},t) = \nabla(\boldsymbol{\eta}(\boldsymbol{r},t)\sqrt{n(\boldsymbol{r},t)}) \tag{2.33}$$

with the global white noise field $\boldsymbol{\eta}$ characterised by

$$\langle \eta^{\mu}(\boldsymbol{r},t)\eta^{\nu}(\boldsymbol{r'},t')\rangle = 2D\delta^{\mu\nu}\delta(t-t')\delta(\boldsymbol{r}-\boldsymbol{r'}), \quad \langle \boldsymbol{\eta}(\boldsymbol{r},t)\rangle = 0. \tag{2.34}$$

The global noise field ξ' has the same correlation function as ξ making them statistically identical and allowing us to rewrite equation 2.29 as

$$\begin{aligned}\frac{\partial n(\boldsymbol{r},t)}{\partial t} = \nabla \cdot \Bigg(& p(\boldsymbol{r},t)\nabla \boldsymbol{\Phi}(\boldsymbol{r_i},t) + D\nabla n(\boldsymbol{r},t) + \sqrt{n(\boldsymbol{r},t)}\boldsymbol{\eta}(\boldsymbol{r},t) \\ & + n(\boldsymbol{r},t)\int \mathrm{d}^3 r'\, n(\boldsymbol{r'},t)\nabla V(|\boldsymbol{r}-\boldsymbol{r'}|)\Bigg).\end{aligned} \tag{2.35}$$

This equation is not mathematically well defined because the solution should be considered a distribution and the square root of a distribution is not always well defined. It is a good representation of the physical model and through spatial discretization a well defined mesoscopic description can be obtained [15]. Equation 2.35 is essentially a more general version of equation 2.6 with an additional noise term. Relating back to the *Electrokinetics* model where we have a electrostatic interaction and an additional advective flux term gives us

$$\begin{aligned}\frac{\partial n_k}{\partial t} &= -\nabla \cdot \boldsymbol{j}_k, \\ \boldsymbol{j}_k &= -D_k\nabla n_k - \nu_k q_k n_k \nabla \Phi - \sqrt{n_k}\boldsymbol{\eta_k} + n_k \boldsymbol{v},\end{aligned} \tag{2.36}$$

with diffusion coefficient D_k, mobility v_k, the electrostatic potential Φ and advective velocity $\boldsymbol{v}$. Together with the in chapter 2 derivated equations for the electrostatic potential and the advective velocity this completes our model.

Chapter 3

Discretization

As shown in chapter 2 the evolution of the particle densities n_k is described by

$$\begin{aligned} \frac{\partial n_k}{\partial t} &= -\nabla \cdot \boldsymbol{j}_k, \\ \boldsymbol{j}_k &= \underbrace{-D_k \nabla n_k}_{j^{\text{diff}}} \underbrace{-\nu_k q_k n_k \nabla \Phi}_{j^{\text{pot}}} \underbrace{-\sqrt{n_k}\eta_k}_{j^{\text{fluc}}} \underbrace{+n_k \boldsymbol{v}}_{j^{\text{adv}}}. \end{aligned} \tag{3.1}$$

We want to discretize the equations 3.1 using a regular cubic grid like it was done for the electrostatic and the advective velocity. In our discretization a grid node represents a cubic cell with uniform density reaching half way to its neighbours. The flux is defined on staggered grid, shifted by $a/2$ compared to the density grid, where a is the distance between grid nodes. Rempfer [24] already implemented a discretization utilizing finite differences and the Gauss' theorem. We choose an alternative approach to make sure the discretization is compatible with the additional noise term. We will derive an 19-point stencil following [22]. We start by considering the problem in one dimension. Approximating ∇j_i with the finite difference method allows us to write

$$\nabla \cdot j_i \approx \frac{1}{a}(j_{i+1/2} - j_{i-1/2}), \tag{3.2}$$

where $j_{i\pm1/2}$ represents the flux on the face between node i with density n_i and node $i \pm 1$ with density $n_{i\pm1}$. We now look at the different contributions to the flux starting with the fluctuations

$$\begin{aligned} j^{\text{fluc}}_{k,i} &= \sqrt{n_{k,i}}\eta_{k,i}, \\ \nabla \cdot j^{\text{fluc}}_{k,i} &\approx \frac{1}{a}(j_{i+1/2} - j_{i-1/2}) \\ &\approx \frac{1}{a}\sqrt{n_{k,i+1/2}}\,\eta_{k,i+1/2} - \sqrt{n_{k,i-1/2}}\,\eta_{k,i-1/2}. \end{aligned} \tag{3.3}$$

The density on the faces $i \pm 1/2$ have to be approximated since the density is only defined on the nodes. A reasonable solution is to approximate via the arithmetic mean $n_{k,i\pm1/2} =$

$\frac{n_{k,i}+n_{k,i\pm1}}{2}$ which yields

$$\nabla \cdot j^{\text{fluc}}_{k,i} \approx \frac{1}{a}\sqrt{\frac{n_{k,i+1}+n_{k,i}}{2}}\,\eta_{k,i+1/2} - \sqrt{\frac{n_{k,i}+n_{k,i-1}}{2}}\,\eta_{k,i-1/2}. \tag{3.4}$$

The second contribution to the flux becomes

$$\begin{aligned} j^{\text{diff}}_{k,i} &= -D_k \nabla n_k \\ \nabla \cdot j^{\text{diff}}_{k,i} &\approx -\frac{D_k}{a}(\nabla n_{k,i+1/2} - \nabla n_{k,i-1/2}). \end{aligned} \tag{3.5}$$

We again use finite differences to approximate $\nabla n_{k,i\pm1/2}$ which yields

$$\begin{aligned} \nabla n_{k,i-1/2} &\approx \frac{1}{a}(n_i - n_{i-1}) \\ \nabla n_{k,i+1/2} &\approx \frac{1}{a}(n_{i+1} - n_i). \end{aligned} \tag{3.6}$$

Inserting 3.6 into 3.5 gives

$$\nabla \cdot j^{\text{diff}}_{k,i} \approx -\frac{D_k}{a^2}(n_{i+1} + n_{i-1} - 2n_i). \tag{3.7}$$

The contribution by the electrostatic potential is treated in similar fashion yielding

$$\begin{aligned} j^{\text{pot}}_{k,i} &= \nu_k q_k n_k \nabla \Phi \\ \nabla \cdot j^{\text{pot}}_{k,i} &\approx -\frac{1}{a^2}\nu_k q_k \left(\frac{n_{k,i+1}+n_{k,i}}{2} \cdot (\Phi_{i+1} - \Phi_i) + \frac{n_{k,i}+n_{k,i-1}}{2} \cdot (\Phi_{i-1} - \Phi_i) \right). \end{aligned} \tag{3.8}$$

The extension to three dimension is straight forward if we only consider flux through the faces of our cubic cells. For example for the contribution by the electrostatic potential we get

$$\nabla \cdot j^{\text{pot}}_{k,i} \approx -\frac{1}{a^2}\nu_k q_k \sum_{j \in \mathcal{N}_f} \left(\frac{n_{k,i}+n_{k,j}}{2} \cdot (\Phi_i - \Phi_j) \right). \tag{3.9}$$

where $\sum_{j\in\mathcal{N}_f}$ is the sum over all nearest neighbours (faces). Only having flux between the nearest neighbours can lead to anisotropies in the dynamics caused by the geometry of the lattice. To avoid this problem we need a stencil that includes the next nearest neighbours as well. In three dimensions we have 6 nearest neighbours (faces) and 12 next nearest neighbours (edges). We start by constructing three orthogonal basis sets that together include all 18 links. For each of the basis sets shown in figure 3.1 we can utilize the same discretization method used to reach equation 3.9.

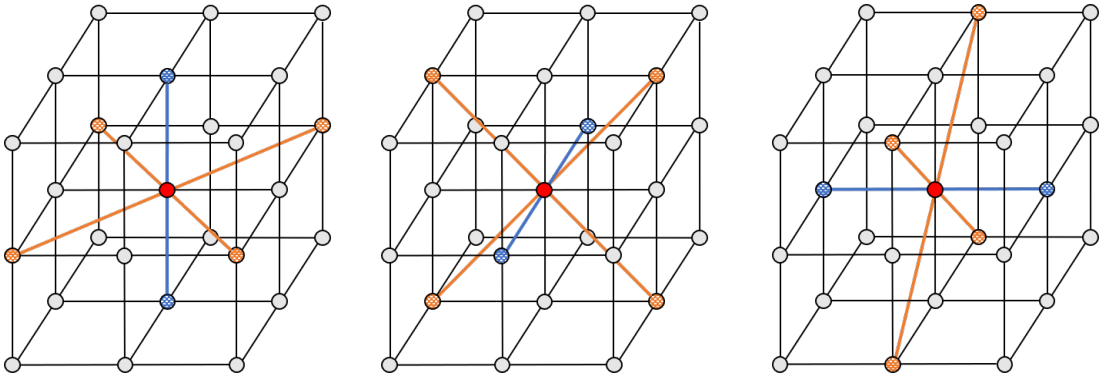

Figure 3.1: Three orthogonal basis sets that together include all 6 faces (blue) and 12 edges (orange).

Taking the average over all three sets then gives us

$$\nabla \cdot j_{k,i}^{\text{pot}} \approx -\frac{1}{3a^2}\nu_k q_k \left(\sum_{j \in \mathcal{N}_f} \frac{n_{k,i} + n_{k,j}}{2} \cdot (\Phi_i - \Phi_j) + \frac{1}{2} \sum_{j \in \mathcal{N}_e} \frac{n_{k,i} + n_{k,j}}{2} \cdot (\Phi_i - \Phi_j) \right) \tag{3.10}$$

$$\nabla \cdot j_{k,i}^{\text{diff}} \approx -\frac{D_k}{3a^2} \left(\sum_{j \in \mathcal{N}_f} (n_{k,j} - n_{k,i}) + \frac{1}{2} \sum_{j \in \mathcal{N}_e} (n_{k,j} - n_{k,i}) \right) \tag{3.11}$$

$$\nabla \cdot j_{k,i}^{\text{fluc}} \approx \frac{\sqrt{2D}}{3a} \left(\sum_{j \in \mathcal{N}_f} \sqrt{\frac{n_{k,i+1} + n_{k,i}}{2}}\, \mathcal{W}_{i \leftrightarrow j} + \frac{1}{\sqrt{2}} \sum_{j \in \mathcal{N}_e} \sqrt{\frac{n_{k,i+1} + n_{k,i}}{2}}\, \mathcal{W}_{i \leftrightarrow j} \right). \tag{3.12}$$

Here $\boldsymbol{\mathcal{W}}$ is a uncorrelated Gaussian random number field defined through $\boldsymbol{\eta} = \sqrt{2D}\boldsymbol{\mathcal{W}}$ with a random number $\boldsymbol{\mathcal{W}}_{i \leftrightarrow j}$ for every link. The first sum in each equation goes over all faces and the second over all edges, the additional factor in front of the second sum comes from the normalisation since the distance to the next nearest neighbours is $\sqrt{2}a$.
We use an explicit first order time stepping scheme and consider that the time integral over white noise is $\int_t^{t+\Delta t} \mathcal{W}(t')_{i \leftrightarrow j} dt' = \sqrt{\Delta t} W_{i \leftrightarrow j}$ which yields

$$n_{k,i}(t + \Delta t) = n_{k,i}(t) - \Delta t \cdot \nabla (j_{k,i}^{\text{pot}} + j_{k,i}^{\text{diff}} + \frac{1}{\sqrt{\Delta t}} j_{k,i}^{\text{fluc}}). \tag{3.13}$$

Considering only j^{diff} the second moment of a particle distribution initially located at node i, after one time step, should match the correct continuum value. The continuum solution for a point source in one dimension is given by

$$n(x,t) = \frac{1}{(4\pi D t)^{\frac{3}{2}}} \exp\left(-\frac{x^2}{4Dt} \right). \tag{3.14}$$

See chapter 5.1 for a short derivation. To calculate the second moment we make use of

$$\int_{-\infty}^{\infty} x^2 e^{-x^2} dx = \frac{1}{2} \int_{-\infty}^{\infty} e^{-x^2} dx - \frac{1}{2} [e^{-x^2}]_{-\infty}^{\infty} = \frac{\sqrt{\pi}}{2}. \tag{3.15}$$

Applying this to the second moment of the particle distribution yields

$$\int_{-\infty}^{\infty} x^2 n(x,t)dx = 2Dt. \tag{3.16}$$

The calculation can easily be extended to d dimensions and assuming isotropic diffusion the second moment is

$$\int_{-\infty}^{\infty} \boldsymbol{r}^2 n(\boldsymbol{r},t)d\boldsymbol{r} = 2dDt. \tag{3.17}$$

To match this value our model has to fulfil

$$\sum_{j\in\mathcal{N}_f} a^2 n_j(\Delta t) + \sum_{j\in\mathcal{N}_e} (\sqrt{2}a)^2 n_j(\Delta t) = 2dD\Delta t, \tag{3.18}$$

where d is the dimensionality. Considering equations 3.10-3.13 this is satisfied since

$$\Delta t \left(6 \cdot a^2 \frac{D}{3a^2} + 12 \cdot (\sqrt{2}a)^2 \frac{D}{6a^2}\right) = 6D\Delta t. \tag{3.19}$$

The last missing part is the advective flux. We rely on the assumption that the density of particles in the volume represented by a node is homogeneous. The advective velocity displaces this volume and transfers it to neighbouring cells. Figure 3.2 shows the scheme for a two dimensional grid.

We finish this chapter with a comparison of the derived discretization with the one initially implemented by Rempfer which is also a 19 point stencil given by

$$\begin{aligned} n_k(\boldsymbol{r},t+\Delta t) &= n_k(\boldsymbol{r},t) - \Delta t \cdot \frac{1}{1+2\sqrt{2}} \sum_i j_{ki}(\boldsymbol{r},t), \\ j_{ki}(\boldsymbol{r},t) &= D_k \cdot \frac{n_k(\boldsymbol{r},t) - n_k(\boldsymbol{r}+\boldsymbol{c}_i,t)}{|\boldsymbol{c}_i|} + \nu_k\, q_k \cdot \frac{n_k(\boldsymbol{r}+\boldsymbol{c}_i,t) + n_k(\boldsymbol{r},t)}{2} \cdot \frac{\Phi(\boldsymbol{r}+\boldsymbol{c}_i) - \Phi(\boldsymbol{r})}{|\boldsymbol{c}_i|}, \end{aligned} \tag{3.20}$$

where $\boldsymbol{r}+\boldsymbol{c}_i$ is the location of the neighbouring node. Writing our new discretization in this notation neglecting the noise term for the moment gives

$$\begin{aligned} n_k(\boldsymbol{r},t+\Delta t) &= n_k(\boldsymbol{r},t) - \Delta t \cdot \frac{1}{3} \sum_i j_{ki}(\boldsymbol{r},t), \\ j_{ki}(\boldsymbol{r},t) &= D_k \cdot \frac{n_k(\boldsymbol{r},t) - n_k(\boldsymbol{r}+\boldsymbol{c}_i,t)}{|\boldsymbol{c}_i|^2} + \nu_k\, q_k \cdot \frac{n_k(\boldsymbol{r}+\boldsymbol{c}_i,t) + n_k(\boldsymbol{r},t)}{2} \cdot \frac{\Phi(\boldsymbol{r}+\boldsymbol{c}_i) - \Phi(\boldsymbol{r})}{|\boldsymbol{c}_i|^2}. \end{aligned} \tag{3.21}$$

While both discretizations fulfill equation 3.18 the normalization is different. In the following section we verify the model and its discretization against a few simple system.

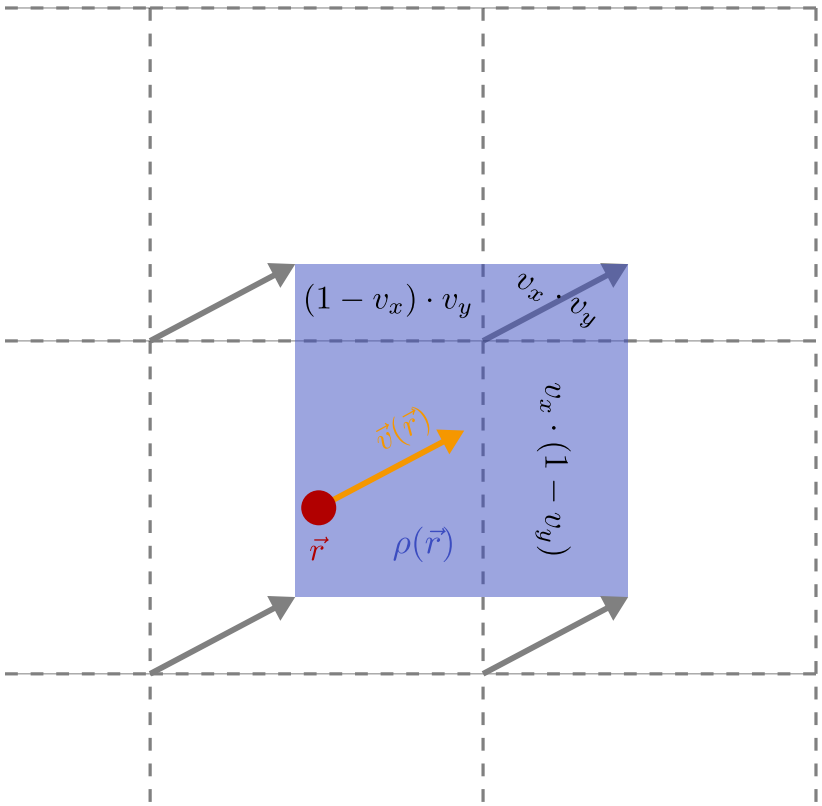

Figure 3.2: Scheme for the advection on a two dimensional grid with spacing h. The dashed lines indicate the boundaries between cells with the lattice nodes in their centre. The flux to the right neighbour would be given by $n(\boldsymbol{r})v_x(1 - v_y)\frac{h^2}{\Delta t}$. Image taken from rempfer [24].

Chapter 4

Implementation

4.1 Compute Unified Device Architecture

The implementation of the *Electrokinetics* feature utilizes the computational power of **G**raphics **P**rocessing **U**nits using the NVIDIA **C**ompute **U**nified **D**evice **A**rchitecture [21]. While the program is still executed on the CPU tasks can be offloaded to the GPU giving a massive performance improvement. A CUDA capable GPU is a vector processor consisting of several **S**treaming **M**ultiprocessors, which each having multiple cores. The cores of one SM execute the same instructions with each core working with a different set of data. Modern GPUs have a huge number of cores representing strong computational power. One bottleneck is often the data transfer to and from the device which is done over the **P**eripheral **C**omponent **I**nterconnect **E**xpress bus. NVIDIA makes use of streamed processing, where tasks, memory transfers and calculations are ordered in streams for subsequent execution. These streams are processed asynchronously to the host programm meaning calculations in one stream can be overlapped by calculations on the CPU and other streams. In order to avoid having a SM being idle while waiting for data multi-threading is used. This allows the SM to execute a different set of threads instead of waiting on the GPU memory, .

The GPU lattice boltzmann method we utilize was implemented in ESPResSo by Röhm [26]. This thesis is built on the implementation of the Electrokinetic solver done by Rempfer [24]. We shortly summarize the basic integration loop for the Electrokinetics Solver and look closer on the newly added fluctuations afterwards.

4.2 Electrokinetics Solver

The integration loop of the Electrokinetics Solver is:

1. Calculate fluxes and propagate the particle densities for each species.
   (a) Clear the fluxes from the previous loop.
   (b) Calculate the fluxes for the current species.
   (c) Clear all fluxes connecting boundary nodes.
   (d) Propagate densities according to the calculated fluxes.
2. Calculate the electrostatic potential
3. Integrate the LB

For details on the implementation and usage of CUDA see Rempfer [24]. We will focus on the newly implemented fluctuations. CUDA is utilized to create the normal random numbers fast, the random number generator is initialized by

```
1  __global__ void initialize_random_generator(int seed) {
2    int index = ek_getThreadIndex();
3    /* Each thread gets same seed, a different sequence ↩
        number, no offset */
4    curand_init(seed, index, 0, &ek_parameters_gpu.↩
        rnd_state[index]);
5  }
```

```
1  auto seed = random_integral<unsigned long long>();
2  KERNELCALL( initialize_random_generator, dim_grid, ↩
      threads_per_block, (seed) );
```

The function adding the fluctuations to the flux is called for every node and loops over the neighbours:

```
1  __device__ void ek_add_fluctuations_to_flux(unsigned int↩
       index, unsigned int species_index, unsigned int *↩
     neighborindex, LB_nodes_gpu lb_node) {
2
3    for(int i = 0; i < 9; i++) {
4      float random = (curand_normal(&ek_parameters_gpu.↩
         rnd_state[index]));
5    .
6    .
```

A direct implementation of Eq. 3.21 allows for fluxes out of an already empty cell since the fluctuation term of the flux depends on the square root of the mean density of the cell and its neighbour. This can lead to negative densities, which are unphysical and our fluctuation flux becomes undefined since it depends on the square root of the densities. The simplest solution to this problem would be turning off the fluctuations for any cell with zero or smaller density, but this would lead to a delta peak in the density distribution at $n = 0$ [15] for simulations with small densities. Instead we opt to use a smoothed Heaviside function defined by

$$H(x) = \begin{cases} 0, & (x \leq 0) \\ x, & (0 < x < 1) \\ 1, & (x \geq 1) \end{cases} \tag{4.1}$$

to modify the arithmetic mean yielding

$$\tilde{n}(n_1, n_2) = \frac{n_1 + n_2}{2} H(n_1) H(n_2). \tag{4.2}$$

This avoids the discontinuity at $n = 0$, while still preventing negative densities. This is easily implemented as

```
12      float H = (density >= 0.0f) * min(density, 1.0f);
13      float HN = (neighbor_density >= 0.0f) * min(↩
          neighbor_density, 1.0f);
14      float mean_density = H * HN * (density+↩
          neighbor_density)/2.0f;
```

We need to differentiate between fluxes to nearest and next nearest neighbours (see equation 3.21)

```
15      //Next nearest neighbours
16      if(i > 2) {
17        float fluc = powf(2.0f * mean_density * D * dt / (↩
              agrid * agrid), 0.5f) * random / (sqrt(3.0f) * ↩
              sqrt(2.0f));
18        fluc *= !(lb_node.boundary[index] || lb_node.↩
              boundary[neighborindex[i]]);
19        flux[jindex_getByRhoLinear(index, i)] += fluc;
20        }
21      //Nearest neighbours
22      else {
23        float fluc = powf(2.0f * mean_density * D * dt / (↩
              agrid * agrid), 0.5f) * random / sqrt(3.0f);
24        fluc *= !(lb_node.boundary[index] || lb_node.↩
              boundary[neighborindex[i]]);
25        flux[jindex_getByRhoLinear(index, i)] += fluc;
26        }
```

Chapter 5

Validation

5.1 Diffusion

We look at the diffusion of a point source and verify that the implemented *Electrokinetics* model and the chosen discretization behaves correctly. We also compare the results with the previously implemented discretization. The system is a number density n_0 initially at $x = 0$ in one dimension. Mass conversion gives us

$$\frac{\partial n}{\partial t} = D\frac{\partial^2 n}{\partial x^2}, \tag{5.1}$$

with the diffusion coefficient D. A solution for $n(x,t)$ can be found through dimensional analysis. $n(x,t)$ can be assumed to be a function of $D\,[m^2 s^{-1}]$, $x\,[m]$, $t\,[s]$ and $n\,[L^{-1}]$. These parameters can form two dimensionless groups and thus yields

$$n\sqrt{Dt} = f(\frac{x}{\sqrt{Dt}}) \tag{5.2}$$

We continue by defining

$$\eta = \frac{x}{\sqrt{Dt}}, \quad \frac{\partial \eta}{\partial t} = -\frac{\eta}{2t}, \quad \frac{\partial \eta}{\partial x} = \frac{1}{\sqrt{Dt}}. \tag{5.3}$$

From here we can calculate

$$\frac{\partial n}{\partial t} = -\frac{n_0}{2t\sqrt{Dt}}\left(f + \eta\frac{\partial f}{\partial \eta}\right), \tag{5.4}$$

$$\frac{\partial^2 n}{\partial x^2} = -\frac{n_0}{2t\sqrt{Dt}}\left(f + \eta\frac{\partial f}{\partial \eta}\right). \tag{5.5}$$

Substituting these equations into 5.1 yields

$$\frac{d^2 f}{d\eta^2} + \frac{1}{2}\left(f + \eta\frac{df}{d\eta}\right) = 0, \tag{5.6}$$

which can solved with

$$f = C\exp\left(-\frac{\eta^2}{4}\right), \tag{5.7}$$

where C can be found considering mass conservation

$$n_0 = \int_V n(x,t)dV \tag{5.8}$$

yielding

$$\int C\exp(-\frac{\eta}{4})d\eta = 1. \tag{5.9}$$

This is solved by $C = \frac{1}{\sqrt{4\pi}}$ and with that we arrive at

$$f = \frac{1}{\sqrt{4\pi}}\exp(\frac{\eta^2}{4}) \tag{5.10}$$

$$n(x,t) = \frac{n_0}{\sqrt{4\pi Dt}}\exp\left(-\frac{x^2}{4Dt}\right). \tag{5.11}$$

In three dimensions with initial condition $n(t=0) = n_0\delta(x)\delta(y)\delta(z)$ and $D_x = D_y = D_z$ the same methods can be applied yielding

$$n(x,t) = \frac{n_0}{(4\pi Dt)^{\frac{3}{2}}}\exp\left(-\frac{x^2+y^2+z^2}{4Dt}\right). \tag{5.12}$$

We simulate a similar system using the *Electrokinetics* model by initializing a single node with density $n_0 = 8$ in an otherwise empty system. We measure the density along the X-axis and along the diagonal of the XY plane (see Fig. 5.1). The diffusion coefficient is set to 1.0, fluctuations are switched off and the system size is chosen to be $L_{x,y,z} = 40$. The simulation was run for 2000 time steps with $\Delta t = 0.01$. As can be seen in Fig. 5.2 and 5.3 there is no significant difference between the two discretization stencils and both show good agreement with the analytical solution for the point source. The small deviations close to the source location are expected since our simulated system is not an actual infinitesimal point source.

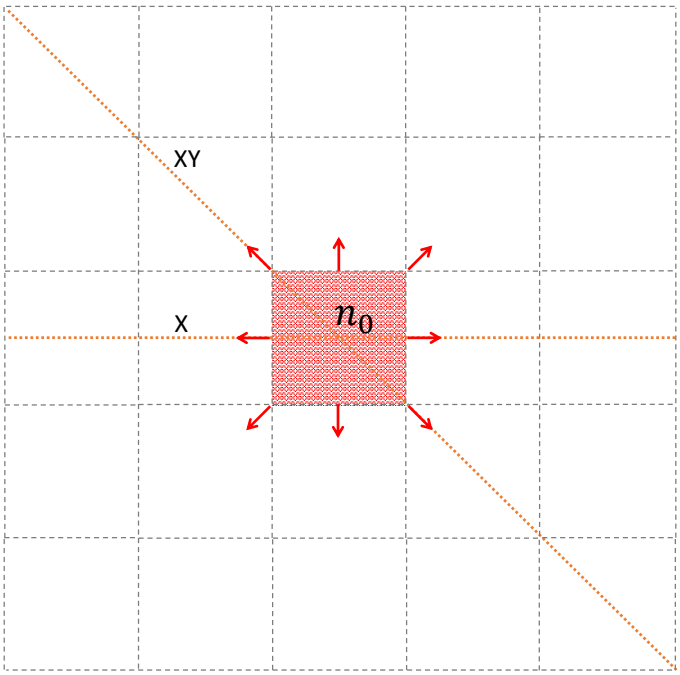

Figure 5.1: 2D sketch of a system initialised with density n_0 in one cell. Orange lines indicate the two axes where values were taken from.

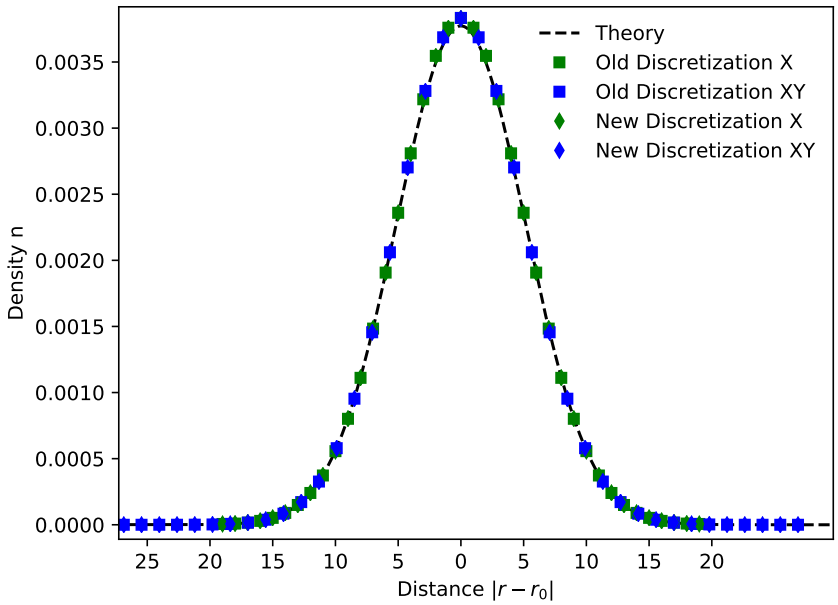

Figure 5.2: Number density n after 200 time steps with $\Delta t = 0.01[t]$ plotted against the distance to initial source location r_0. Distance is given in units of the lattice parameter. Values for the density were taken along the X-axis and along the diagonal of the XY plane.

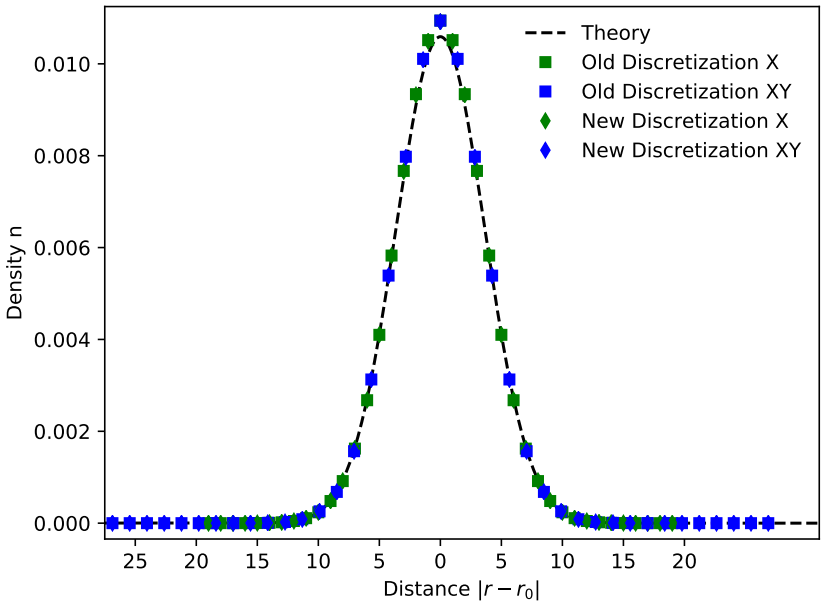

Figure 5.3: Number density n after 100 time steps with $\Delta t = 0.01[t]$ plotted against the distance to initial source location r_0. Distance is given in units of the lattice parameter. Values for the density were taken along the X-axis and along the diagonal of the XY plane.

5.2 Ideal Gas

The simplest system to verify the fluctuations of the continuum model described and implemented in the previous chapters is an ideal gas. Considering a system with volume V containing N uncharged non-interacting particles we can find the probability $P_N(n)$ of a partial volume v to contain n particles as

$$P_N(n) = \binom{N}{n} p^n (1-p)^{N-n}, \tag{5.13}$$

where p is the probability for a particle to be in v. For our system p is simply the volume ratio $\frac{v}{V}$ which yields

$$P_N(n) = \frac{N!}{n!(N-n)!} \left(\frac{v}{V}\right)^n \left(\frac{V-v}{V}\right)^{N-n}. \tag{5.14}$$

For big systems where $V >> v$ and $N \to \infty$ Eq. 5.13 simplifies to the *Poisson* distribution

$$P(n) = \frac{\bar{n}^n e^{-\bar{n}}}{n!}, \tag{5.15}$$

with the mean $\bar{n} = N\frac{v}{V}$. Using *Stirling's approximation* $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ yields the continuous function

$$P(n) = \frac{1}{\sqrt{2\pi n}}\left(\frac{\bar{n}}{n}\right)^n e^{n-\bar{n}}. \tag{5.16}$$

Equations 5.13 - 5.16 all do not depend on the temperature or mobility of the particles. The approximated density distribution $P(n)$ in Eq. 5.16 only depends on the mean particle Number $\bar{n}$.

A simulation of a system containing a single uncharged species and no background fluid is set up with the *Electrokinetics* feature of ESPResSo. The system has periodic boundary conditions and the box is chosen to be a cube with $L = 18[x]$. All interactions with the *Lattice-Boltzmann* fluid are disabled in order to create an ideal gas like system. The number density of the species is chosen to be $27/[m]^3$, the diffusion constant is set to $1.0\frac{[x]^2}{[t]}$ and the time step is $\Delta t = 0.0005\,[s]$. The simulation is run for 10^7 integration steps and the number of particles in every cell is saved every 1000 steps. Figure 5.4 shows the histogram of the density data of all cells and time steps.

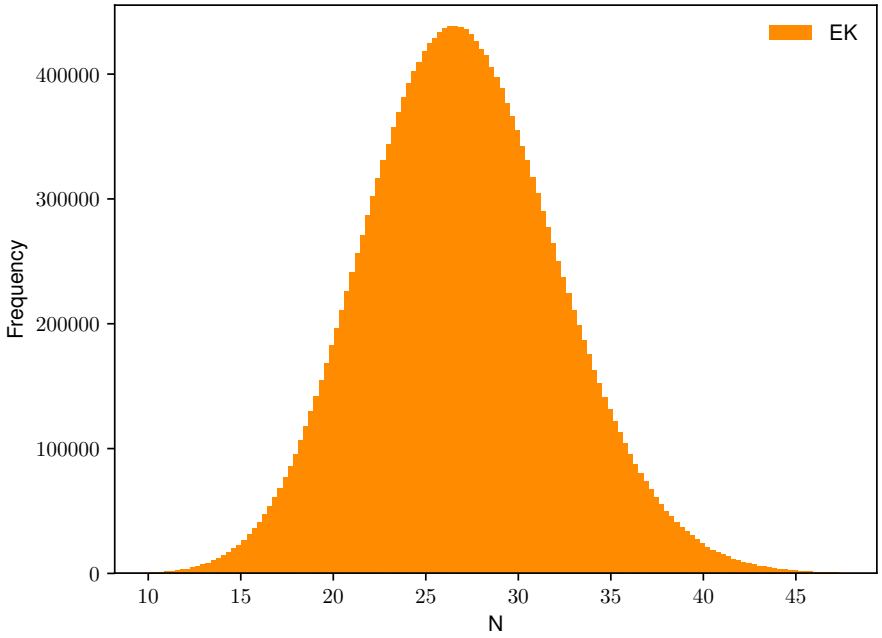

Figure 5.4: Frequency of different values for the number density. Bin size is 0.275, with the first bin starting at $N = 10$. The system is an ideal gas simulated using the *Electrokinetics* model.

In addition to the simulation using the *Electrokinetics* feature a simulation using explicit

point particles was done. For this plain molecular dynamics simulation a langevin thermostat with friction $\gamma = 1.0$ and $k_\mathrm{b}T = 1.0\,[E]$ is used to thermalise the particles. Other simulation parameters are chosen to be the same as in the *Electrokinetics* simulation. This MD system consisting of particles undergoing independent Brownian motion is the system we discussed for the derivation of the noise term in chapter 2.4 making it the obvious choice to validate our implementation of the continuum model. For the analysis the simulation box is divided into cells and every 1000 steps the number of particles in every cell is counted and saved.

In figure 5.5 we can see that the *Electrokinetics* and the molecular dynamics simulation are both replicating the expected distribution (Eq. 5.16) of an ideal gas.

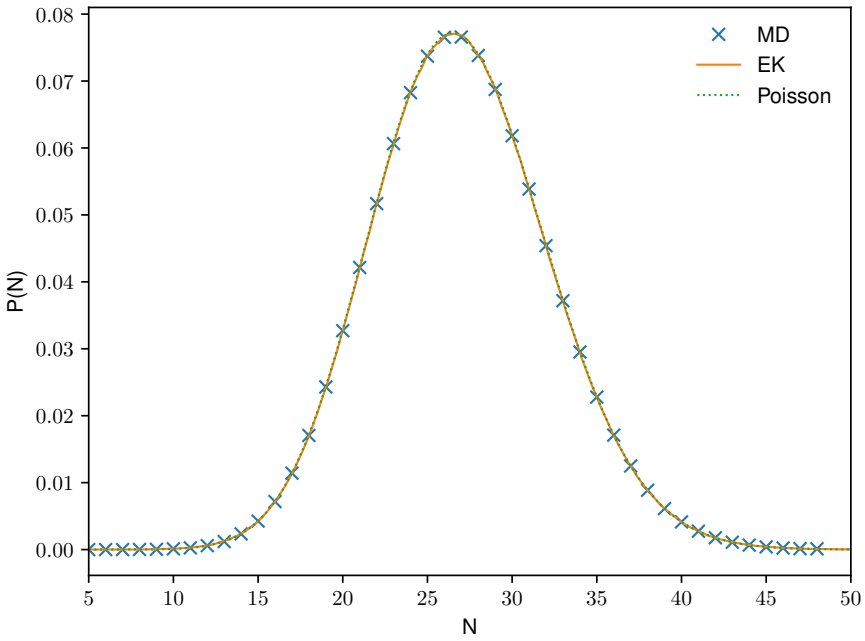

Figure 5.5: Comparison of the number density distribution for the *Electrokinetics* simulation (orange), the *MD* simulation (blue) and the theoretical solution (green).

5.3 Coloumb Gas

We now look at a more complex system containing two oppositely charged ion species. The two species have a charge of $q_1 = +1[q]$ and $q_2 = -1[q]$ respectively. The number density is the same for both and is set to $108/[m]^3$, which means the system in total is neutral. The other simulation parameters are chosen to be the same as in the Ideal gas simulation.

We again compare the results with a molecular dynamics simulation. The electrostatic interactions in the MD system are handled with the P3M algorithm implemented in ESPResSo and we set the Bjerrum length to $l_{\mathrm{B}} = 1.0$. In order to prevent a collapse into positive/negative pairs a *Lennard Jones* potential is added to the particles. The potential is defined by [11]

$$V_{\mathrm{LJ}}(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r - r_{\mathrm{off}}} \right)^{12} - \left(\frac{\sigma}{r - r_{\mathrm{off}}} \right)^{6} + c_{\mathrm{shift}} \right] & \text{if } r_{\mathrm{min}} + r_{\mathrm{off}} < r < r_{\mathrm{cut}} + r_{\mathrm{off}} \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

where ϵ is the depth of the potential well, σ is the distance at which the potential is zero, r is the distance between particles, r_{min} is the distance at which the potential reaches its minimum and beyond r_{cut} the potential is cut off and the interaction force becomes zero. For the simulation we set $\sigma = 0.225[x]$ and $\epsilon = 1.0[E]$. The cutoff is chosen to be $r_{\mathrm{cut}} = 2^{\frac{1}{6}}\sigma$, which yields the purely repulsive *Weeks-Chandler-Andersen* potential.
The resulting density distributions $P(n)$ of the *Electrokinetics* and the molecular dynamics simulation match as shown in figure 5.6.

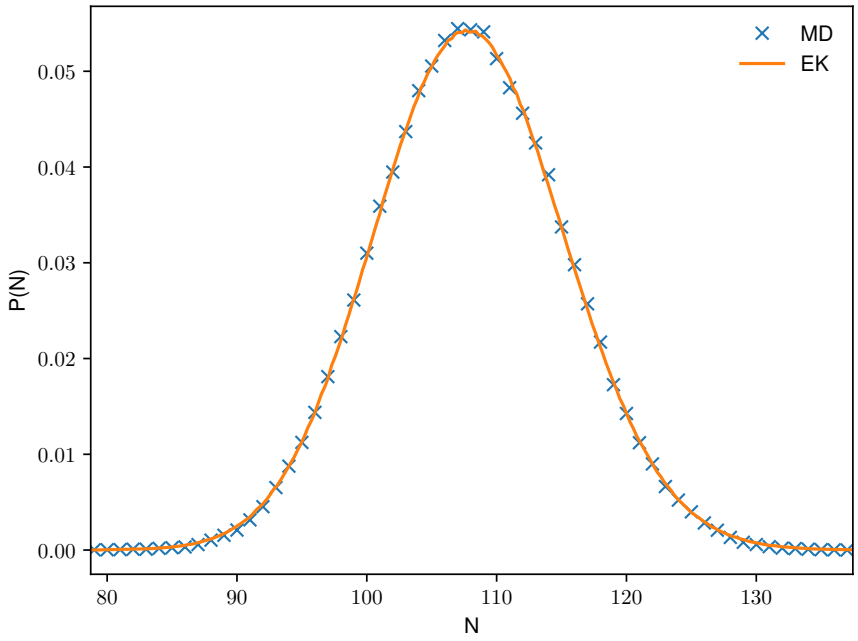

Figure 5.6: Comparison of the number density distribution of a coloumb gas simulated using the *Electrokinetics* model (orange) and a *MD* simulation (blue).

5.4 Spatial Correlations

The second moment of the simulated distribution and the Poisson distribution should also agree. The second moment of the Poisson distribution is given by

$$\langle n(\boldsymbol{r},t)n(\boldsymbol{r},t)\rangle = \sum_{n=0}^{\infty} n\langle n(\boldsymbol{r},t)n(\boldsymbol{r},t)\rangle = \bar{n}\bar{n} + \bar{n}, \tag{5.18}$$

where $\bar{n}$ is the mean particle number. In order to verify the second moment in our simulation we average over T time steps and the entire lattice, this can be expressed by

$$\sum_{t=1}^{T}\sum_{\boldsymbol{r}} n(\boldsymbol{r},t)n(\boldsymbol{r},t) \stackrel{!}{=} TV(\bar{n}\bar{n} + \bar{n}), \tag{5.19}$$

where V the number of lattice nodes in our system. We rewrite this expression as

$$\sum_{t=1}^{T}\sum_{\boldsymbol{r}} \frac{n(\boldsymbol{r},t)n(\boldsymbol{r},t)}{TV(\bar{n}\bar{n} + \bar{n})} \stackrel{!}{=} 1. \tag{5.20}$$

Evaluating this for the data of the Ideal gas simulation (chapter 5.2) yields

$$\sum_{t=1}^{T}\sum_{\boldsymbol{r}} \frac{n(\boldsymbol{r},t)n(\boldsymbol{r},t)}{TV(\bar{n}\bar{n} + \bar{n})} = 0.99352363, \tag{5.21}$$

showing a clear agreement with the anticipated second moment but no exact equality since the Poisson distribution assumes an infinite system. We analyse the second moment further and try to separate globally conserved modes from non-conserved modes by making use of Fourier transforms [29]. We assume a square lattice with L nodes in each spatial direction and define

$$n(\boldsymbol{k},t) = \sum_{\boldsymbol{r}} \exp\left(\frac{2\pi}{L}\boldsymbol{k}\cdot\boldsymbol{r}\right) n(\boldsymbol{r},t). \tag{5.22}$$

A structure factor for the densities in a three dimensional system can be written as

$$\begin{aligned} S(\boldsymbol{k}) = \langle n(\boldsymbol{k},t)n(-\boldsymbol{k},t)\rangle &= \sum_{\boldsymbol{r}}\sum_{\boldsymbol{r'}} \exp\left(\frac{2\pi}{L}\boldsymbol{k}\cdot(\boldsymbol{r}-\boldsymbol{r'})\right)\langle n(\boldsymbol{r},t)n(\boldsymbol{r'},t)\rangle \\ &= \delta_{k0}L^6\bar{n}\bar{n} + L^3\bar{n}. \end{aligned} \tag{5.23}$$

For $S(\mathbf{0})$ Mass conversation imposes

$$n(\mathbf{0},t)n(\mathbf{0},t) = L^6\bar{n}^2. \tag{5.24}$$

We therefore change our equation for $S(\boldsymbol{k})$ to respect Mass conservation and predict

$$S^{\mathrm{Th}}(\boldsymbol{k}) = \delta_{k0}L^6\bar{n}\bar{n} + (1-\delta_{k0})L^3\bar{n}. \tag{5.25}$$

We expect our model to satisfy this but we predict small k-dependent deviations of the Fourier modes caused by the discretization [9]. The deviation

$$S^{\mathrm{Dif}}(\boldsymbol{k}) = \frac{\langle n(\boldsymbol{k},t)n(-\boldsymbol{k},t)\rangle - S^{\mathrm{Th}}(\boldsymbol{k})}{S^{\mathrm{Th}}(\boldsymbol{k})} \tag{5.26}$$

for a $L = 10$ lattice averaged over 10^6 time steps is shown in Fig. 5.7. The results show deviations between 0.5% and 2.5% with a noticeable $\boldsymbol{k}$ dependence. An important limit

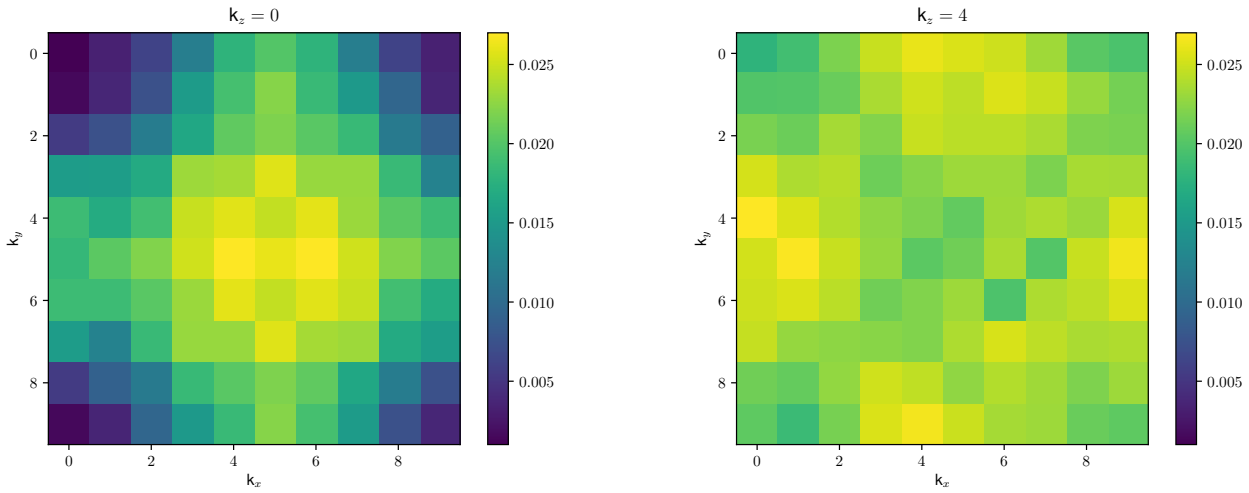

Figure 5.7: Relative deviation of $S(\boldsymbol{k})$ from the theory for a 10x10x10 lattice averaged over 10^6 time steps. There is a noticeable dependence on $\boldsymbol{k}$.

of our model are very small densities because the Poisson distribution will not resemble a continuous distribution anymore and our continuous description of fluctuations will no longer be viable. The in chapter 4 introduced cut-off that prevents negative densities will lead to further errors. We investigate this low density limit of the model by examining the $\langle n(\boldsymbol{r},t)n(\boldsymbol{r},t)\rangle$ for decreasing densities (see Fig. 5.8). The relative deviation from the theory $\langle n(\boldsymbol{r},t)n(\boldsymbol{r},t)\rangle/(\bar{n}^2+n)$ starts to exponentially grow once densities lower than four particles per cell are reached. For $\bar{n} > 5$ the deviation stays under around 0.1% showing very good agreement with second moment of the Poisson distribution.

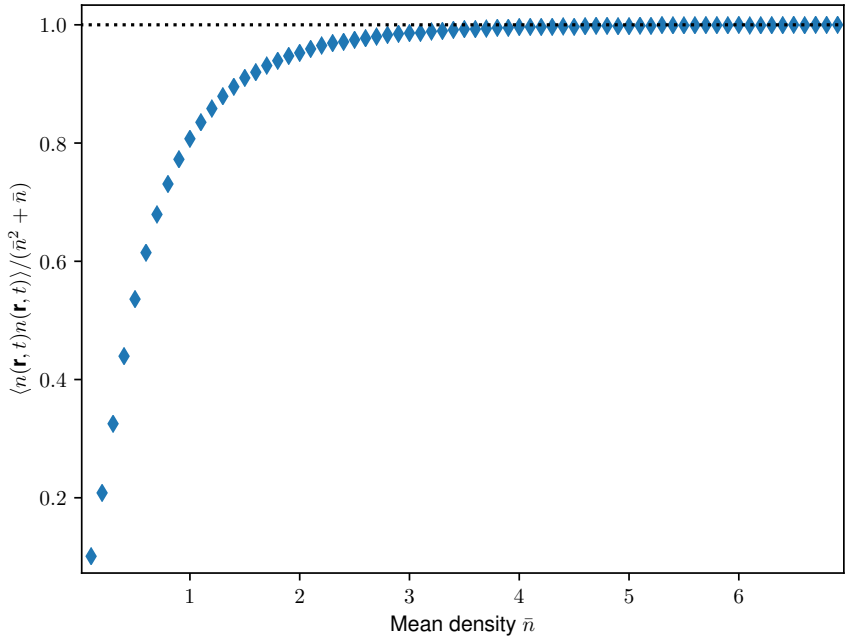

Figure 5.8: $\langle n(\boldsymbol{r},t)n(\boldsymbol{r},t)\rangle/(\bar{n}^2+n)$ for decreasing densities. Values are averaged over 10^4 integration steps and over the $10\times10\times10$ lattice.

5.5 Time correlation

In addition to the spatial correlations we take a look at the time correlations. A function for the time-correlation of our discretization can be derived from the Fourier transformation of the diffusion equation [29]. Starting from the diffusive part of our model, neglecting other contributions, we write

$$n_i(t+1) = n_i(t) - \frac{D}{3}\Big(\sum_{j\in\mathcal{N}_f} (n_j - n_i) + \frac{1}{2}\sum_{j\in\mathcal{N}_e} (n_j - n_i)\Big), \tag{5.27}$$

where we set a, $\Delta t = 1$ for simplicity. Changing the notation from indicating the node via index i to coordinates (x, y, z)

$$\begin{aligned}
&n(x,y,z,t+1) - n(x,y,z,t)\\
&\quad= \frac{D}{3}\big(n(x+1,y,z,t) + n(x,y+1,z,t) + n(x,y,z+1,t)\\
&\qquad +n(x-1,y,z,t) + n(x,y-1,z,t) + n(x,y,z-1,t)\\
&\qquad -6n(x,y,z,t)\big)\\
&\quad+\frac{D}{6}\big(n(x+1,y+1,z,t) + n(x-1,y+1,z,t) + n(x+1,y-1,z,t) + n(x-1,y-1,z,t)\\
&\qquad +n(x,y+1,z+1,t) + n(x,y-1,z+1,t) + n(x,y+1,z-1,t) + n(x,y-1,z-1,t)\\
&\qquad +n(x+1,y,z+1,t) + n(x-1,y,z+1,t) + n(x+1,y,z-1,t) + n(x-1,y,z-1,t)\\
&\qquad -12n(x,y,z,t)\big).
\end{aligned} \tag{5.28}$$

The discrete Fourier transformation of this equation is

$$\begin{aligned}
&\sum_{x,y,z} e^{i2\pi(\frac{k_x x}{L_x} + \frac{k_y y}{L_y x} + \frac{k_z z}{L_z})}[n(x,y,z,t+1) - n(x,y,z,t)]\\
&\quad= \frac{D}{3}\Bigg[e^{i2\pi\frac{k_x}{L_x}} + e^{-i2\pi\frac{k_x}{L_x}} + e^{i2\pi\frac{k_y}{L_y}} + e^{-i2\pi\frac{k_y}{L_y}} + e^{i2\pi\frac{k_z}{L_z}} + e^{-i2\pi\frac{k_z}{L_z}}\\
&\qquad + \frac{1}{2}e^{i2\pi(\frac{k_x}{L_x} + \frac{k_y}{L_y})} + \frac{1}{2}e^{i2\pi(\frac{k_x}{L_x} - \frac{k_y}{L_y})} + \frac{1}{2}e^{-i2\pi(\frac{k_x}{L_x} + \frac{k_y}{L_y})} + \frac{1}{2}e^{-i2\pi(\frac{k_x}{L_x} - \frac{k_y}{L_y})}\\
&\qquad + \ ... \ - 12\Bigg] \sum_{x,y,z} e^{i2\pi(\frac{k_x x}{L_x} + \frac{k_y y}{L_y} + \frac{k_z z}{L_z})} n(x,y,z,t)\\
&\quad=\gamma(k_x,k_y,k_z) \sum_{x,y,z} e^{i2\pi(\frac{k_x x}{L_x} + \frac{k_y y}{L_y} + \frac{k_z z}{L_z})} n(x,y,z,t)
\end{aligned} \tag{5.29}$$

with

$$\begin{aligned}
&\gamma(k_x,k_y,k_z) \\
&= \frac{D}{3}\Bigg[12 - 2\cos\left(\pi\frac{k_x}{L_x}\right) - 2\cos\left(\pi\frac{k_y}{L_y}\right) - 2\cos\left(\pi\frac{k_z}{L_z}\right) \\
&\qquad - \cos\left(\pi(\frac{k_x}{L_x}+\frac{k_y}{L_y})\right) - \cos\left(\pi(\frac{k_x}{L_x}-\frac{k_y}{L_y})\right) - \cos\left(\pi(\frac{k_y}{L_y}+\frac{k_z}{L_z})\right) \\
&\qquad - \cos\left(\pi(\frac{k_y}{L_y}-\frac{k_z}{L_z})\right) - \cos\left(\pi(\frac{k_x}{L_x}+\frac{k_z}{L_z})\right) - \cos\left(\pi(\frac{k_x}{L_x}-\frac{k_z}{L_z})\right)\Bigg]
\end{aligned} \tag{5.30}$$

The solution is given by

$$n(k_x,k_y,k_z,t) = n(k_x,k_y,k_z,0)e^{-\gamma(k_x,k_y,k_z)t}. \tag{5.31}$$

For the time correlation function we therefore expect

$$c(k_x,k_y,k_z,t) = \frac{\langle n(k_x,k_y,k_z,0)n(-k_x,-k_y,-k_z,t)\rangle}{\langle n(k_x,k_y,k_z,0)n(-k_x,-k_y,-k_z,0)\rangle} = e^{-\gamma(k_x,k_y,k_z)}. \tag{5.32}$$

Fig. 5.9 shows good agreement for different Fourier modes. The values were averaged over $3.6 \cdot 10^6$ iterations.

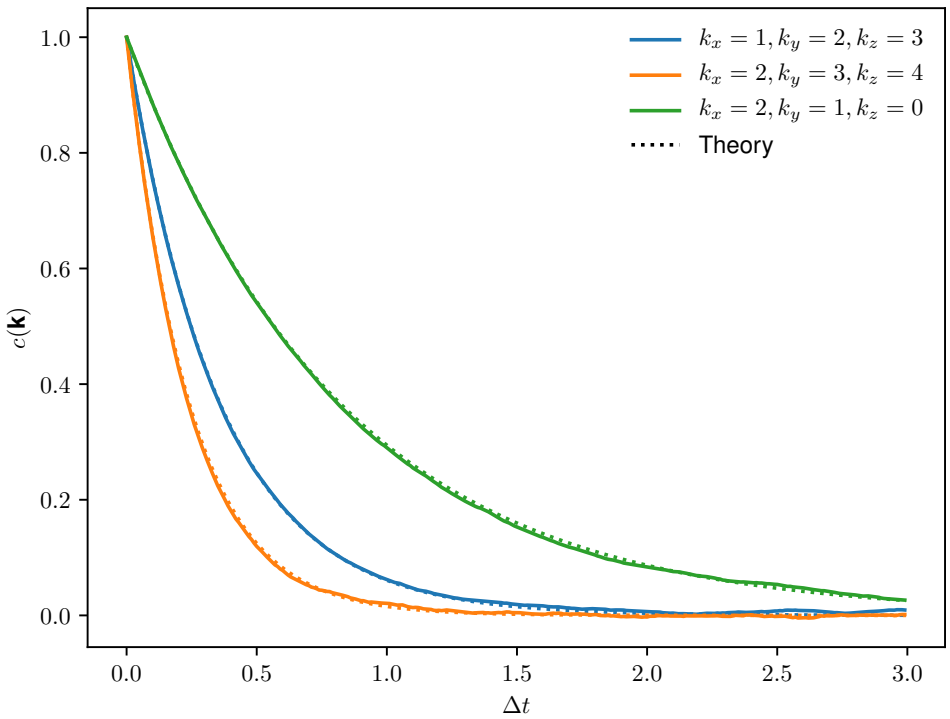

Figure 5.9: Time correlation of Eq. 5.32 for a lattice with $L_x = L_y = L_z = 10$, $D = 1.0$ and mean density $\bar{n} = 120$. The values have been averaged over 10^6 iterations

Chapter 6

Polymer Simulations

In the previous chapters we validated that the new fluctuations behave as expected and results from simulations using the EK model match with theory and MD-Simulations for simple systems such as an ideal gas.
The big advantage of using a continuum model for the ions is that we can simulate systems containing lots of ions cheaper compared to models using explicit ions. With the now thermalised model we can look to simulate complex soft matter systems, where the energy scale of the interactions is in the range of the thermodynamic energy, which means thermal flucutations can play an important role.

The old, not thermalised implementation of the EK model was used by Schoell [27] to simulate the ions in a system consisting of a polyelectrolyte dissolved in water and ions from its autoprotolysis. While the fluid handled by *Lattice Boltzmann* and the particles of the polymer were properly thermalised, the noise in the ion fluxes was missing. The results were compared to Grass et. al. [13][12] who used particle ions to simulate the same physical system. The results showed that while the continuum model is able to simulate big systems with high salt concentrations faster than a particle based simulation, quantitatively the simulation results did not match the correct physical behaviour very well. Schoell proposed two ways to improve the model, an excluded volume interaction between ion species and MD-Particles and the thermalisation of the ion fluxes. With the thermalisation of the ion fluxes now implemented we revisit these polyelectrolyte simulations in the hope of improving the results.

6.1 Model

The system of interest is a polyelectrolyte in water with dissolved ions. The polymer is charged after dissociating counterions into the fluid. We look at the case where only

those counterions are present and no salt is added. The polymer is modelled using a coarse-grained bead-spring model. Each bead can represent a single monomer or several depending on which resolution is needed and which molecule is investigated. The beads interact with each other via a WCA (***W**eeks-**C**handler-**A**ndersen*) potential, an electrostatic potential and a pair-potential. We use the FENE (***f**initely **e**xtensible **n**onlinear **e**lastic*) pair-potential. The fluid and the dissolved ions are handled by the *Electrokinetics* model introduced in previous chapters, whereas the beads are simulated as explicit (**M**olecular **D**ynamics) particles. The beads and ions can interact through electrostatic and hydrodynamic interactions but there is no excluded volume interaction between ions and beads. The thermalisation of the beads/monomers is achieved through the *Lattice-Boltzmann* fluid by fluctuations of the coupling force between fluid and particles. The fluid is thermalised by fluctuations added to the stress tensor and the ions are thermalised with the in chapter 2.4 derived fluctuations of the flux.
Since we later compare our results to Schoell [27] and Grass et. al. [13][12] we list the differences of the used models. The model used by Schoell only differs in the description of the ion species, using the previous discretization (see eq. 3.20) and having no fluctuation term in the ion fluxes.
Grass et. al. uses MD particles for both ions and monomers and *Lattice Boltzmann* for the fluid. The ions and monomers are both thermalised through the coupling with the LB fluid and the fluid is like in our model thermalised through noise applied to the stress tensor. Another notable difference is that there is an excluded volume interaction between ions and monomers in the model used by Grass et. al., which is missing in Schoells and our model.

The WCA potential has the same form as the one used in 5.3 (eq. 5.17)

$$V_{\mathrm{LJ}}(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r - r_{\mathrm{off}}} \right)^{12} - \left(\frac{\sigma}{r - r_{\mathrm{off}}} \right)^{6} + c_{\mathrm{shift}} \right] & \text{if } r_{\mathrm{min}} + r_{\mathrm{off}} < r < r_{\mathrm{cut}} + r_{\mathrm{off}} \\ 0 & \text{otherwise} \end{cases}$$

with the cut-off at $r_{\mathrm{cut}} = 2^{\frac{1}{6}}$.
The FENE potential we use for the bonds between monomers is [11]

$$V_{\mathrm{FENE}}(r) = -\frac{1}{2} K \cdot \Delta r_{\mathrm{max}}^2 \ln \left(1 - \left(\frac{r}{r_{\mathrm{max}}} \right)^2 \right), \tag{6.1}$$

where K is the stiffness of the spring. The beads interact with the fluid via friction and the associated force is [11]

$$\boldsymbol{F}_{\mathrm{hydr.}} = -\gamma(\boldsymbol{v} - \boldsymbol{u}) + \boldsymbol{F}_R, \tag{6.2}$$

where $\boldsymbol{v}$ is the particle velocity, $\boldsymbol{u}$ the local fluid velocity and $\boldsymbol{F}_R$ the random force of the thermalisation. The fluid velocity is interpolated from surrounding cells to the particle position giving us an effective hydrodynamic radius.

The electrostatic interaction between beads is handled with the *Particle-Particle-Particle-Mesh* (P3M) algorithm, which is derived from *Ewald summation*. For a detailed description see Deserno and Holm [6][7].

The *Electrokinetic* ion species are coupled to the MD-Particles via the hydrodynamic interaction mediated by the *Lattice-Boltzmann* fluid and via electrostatic interactions. Figure 6.1 gives an overview over the interactions present in the model.

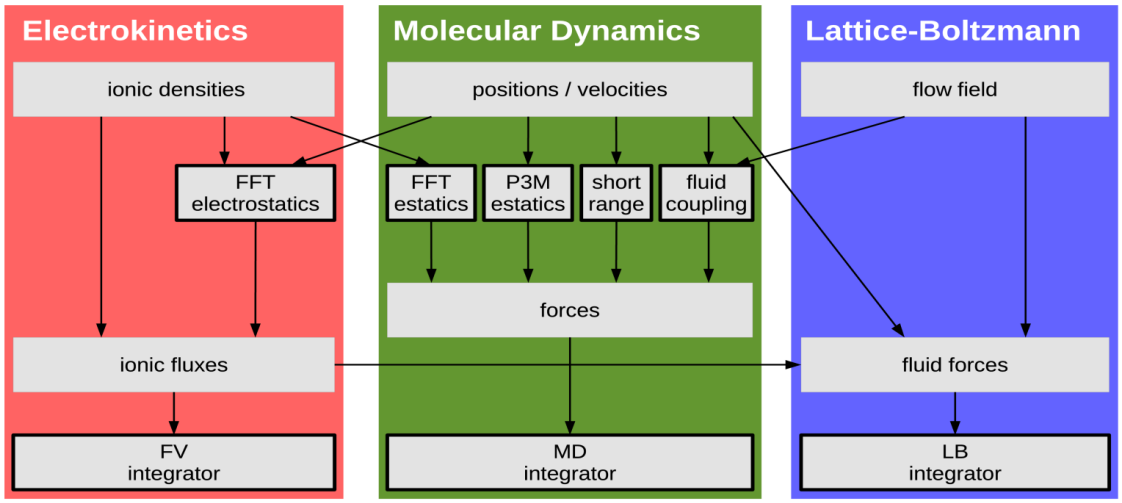

Figure 6.1: Overview over the different interactions present in the model. The advective force of the fluid onto the ions is not shown. Image taken from Schoell [27].

The electrostatic interaction uses the in chapter 2.3 described FFT method to solve the *Poisson's* equation. Since the particles already interact with each other over the P3M algorithm we need to take care that the EK species - particle interaction is handled in a way that does not include the particle-particle interactions a second time. We ensure this by calculating two potentials, for one potential we include the charge of the ions and of the particles

$$\Delta\Phi_{EK+MD}(\boldsymbol{r}) = -4\pi l_{\mathrm{B}} k_{\mathrm{B}} T \sum_k q_k n_k(\boldsymbol{r}) + \sum_i q_i \delta(\boldsymbol{r} - \boldsymbol{r}_i), \tag{6.3}$$

for the other potential we only consider the ions

$$\Delta\Phi_{EK}(\boldsymbol{r}) = -4\pi l_{\mathrm{B}} k_{\mathrm{B}} T \sum_k q_k n_k(\boldsymbol{r}). \tag{6.4}$$

We use the potential including all charges $\Phi_{EK+MD}(\boldsymbol{r})$ to calculate the electrostatic forces on the ions and the other potential $\Phi_{EK}(\boldsymbol{r})$ to calculate the force on the particles.
In order to calculate $\Phi_{EK+MD}(\boldsymbol{r})$ we need to map the charges of the particles to the grid on which the EK species are defined. We first find in which cell a particle is then calculate the position of the particle relative to the center of the cell and then map the charge q to the cell it is in and the surrounding cells keeping the center of the charge distribution at the position of the particle. Figure 6.2 shows the mapping scheme in 2D.

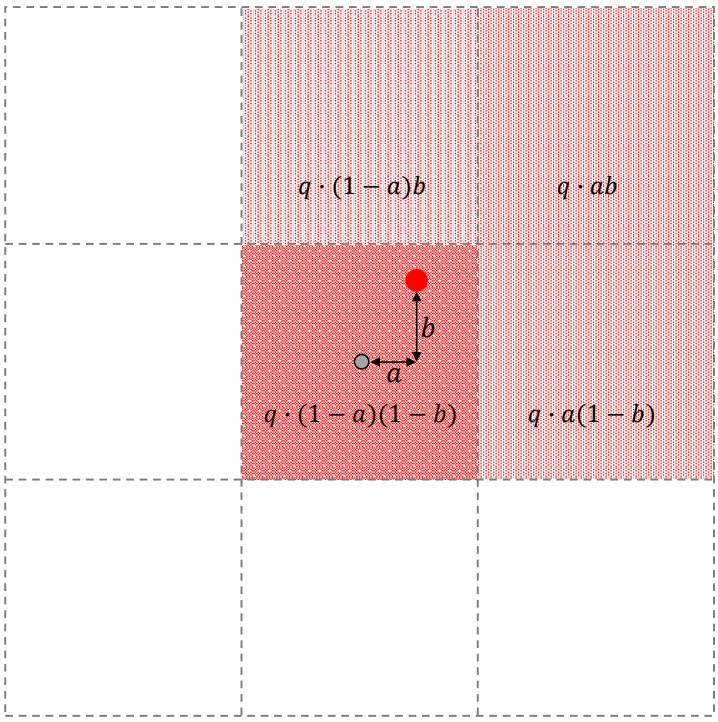

Figure 6.2: 2D Visualization of the scheme used to map the point charges to the lattice of the EK. The charge q (red circle) is distributed among the cells depending on the location of the point charge in relation to the lattice. The grey circle represents the center of the cell and a,b are the distances of the point charge to that center.

We use a *Velocity-Verlet* integration scheme to solve Newton's equations of motion for the beads. The *Velocity-Verlet* can be derived through Taylor expansions:

$$\begin{aligned}
\boldsymbol{r}(t+\Delta t) &= \boldsymbol{r}(t) + \dot{\boldsymbol{r}}(t)\Delta t + \frac{1}{2}\ddot{\boldsymbol{r}}(t)\Delta t^2 + O(\Delta t^3), \\
\boldsymbol{v}(t+\Delta t) &= \boldsymbol{v}(t) + \dot{\boldsymbol{v}}(t)\Delta t + \frac{1}{2}\ddot{\boldsymbol{v}}(t)\Delta t^2 + O(\Delta t^3), \\
\boldsymbol{a}(t+\Delta t) &= \boldsymbol{a}(t) + \dot{\boldsymbol{a}}(t)\Delta t + O(\Delta t^2).
\end{aligned} \tag{6.5}$$

We want to express $\ddot{\boldsymbol{v}}(t)$ in terms of known quantities

$$\Delta t\ddot{\boldsymbol{v}}(t) = \boldsymbol{a}(t+\Delta t) - \boldsymbol{a}(t), \tag{6.6}$$

which allows us to write

$$\boldsymbol{v}(t+\Delta t) = \boldsymbol{v}(t) + \frac{1}{2}\boldsymbol{a}(t)\Delta t + \frac{1}{2}\boldsymbol{a}(t+\Delta t)\Delta t. \tag{6.7}$$

The *Velocity-Verlet* is now given by

$$\begin{aligned}
\boldsymbol{r}(t+\Delta t) &= \boldsymbol{r}(t) + \boldsymbol{v}(t)\Delta t + \frac{\boldsymbol{F}(t)}{2m}\ddot{\boldsymbol{r}}(t)\Delta t^2, \\
\boldsymbol{v}(t+\Delta t) &= \boldsymbol{v}(t) + \frac{\boldsymbol{F}(t)}{2m}\Delta t + \frac{\boldsymbol{F}(t+\Delta t)}{2m}\Delta t,
\end{aligned} \tag{6.8}$$

where we used $\boldsymbol{F} = m \cdot \boldsymbol{a}$. The scheme for the *velocity Verlet* algorithm is then given by:

1. Calculate the new positions $\boldsymbol{r}(t+\Delta)$
2. Calculate the first part of new velocity $\boldsymbol{v}' = \boldsymbol{v}(t) + \frac{F}{2m}\Delta t$
3. Calculate the forces $\boldsymbol{F}(t+\Delta)$ and the new positions $\boldsymbol{r}(t+\Delta)$
4. Calculate the second part of the new velocity
   and add it to the first $\boldsymbol{v}(t+\Delta) = v' + \frac{\boldsymbol{F}(t+\Delta t)}{2m}\Delta t$

For the analysis of the polymer simulation we look at the end-to-end distance R_e, the radius of gyration R_g and the hydrodynamic radius R_h. The end-to-end distance is simply the distance between first and last monomer

$$R_e = |\boldsymbol{r}_1 - \boldsymbol{r}_N|. \tag{6.9}$$

The radius of gyration expresses the moment of inertia and is given by

$$R_g^2 = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{r}_i - \boldsymbol{r}_{com})^2. \tag{6.10}$$

A hydrodynamic radius R_h means the polymer has a comparable flow resistance to a sphere with radius R_h and is calculated with

$$\frac{1}{R_h} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|}. \tag{6.11}$$

These radii fluctuate since the polymer changes conformation over time. We thus are interested in the expectation value of these observables. In order to get an approximation of the true expectation value we run the simulation for a long time and take the time average of the measurements made every integration loop.

Depending on the interactions between monomers and the properties of the solvent a polymer will assume different conformations. Because of thermal fluctuations and the volume of the monomers the polymer will never collapse into a single point, we can estimate the radius of the polymer with

$$R \propto (N-1)^{\nu} \tag{6.12}$$

where ν is the *flory* scaling exponent and N the number of monomers the polymer consist of. The scaling exponent describes how the conformation of a polymer behaves and how the radii such as the end-to-end distance grow with the number of monomers. One differentiates between *bad* solvents with $\nu < 0.588$ and *good* solvents with $\nu >= 0.588$. For *bad* solvents the net interaction between monomers is attractive and the polymer will therefore take on a compact conformation with a small scaling exponent. For a *good* solvent the net interaction is repulsive meaning the polymer will be more stretched out with a high scaling exponent. For the system we chose we expect the behaviour of a good solvent because the charged monomers will have a strong repulsive interaction with no additional salt present.

6.2 Results

We repeat the polymer simulations done by Schoell [27] with the now thermalised EK model. We simulate polymers with a length of 3-50 monomers, at a constant monomer concentration of 5 mM. Each monomer has a charge of $q = 1$ apart from the counterions neutralizing the system there are no extra salt ions present. We chose the same diffusion coefficient as Schoell $D = 6.51472 \cdot 10^{-4}\,[x^2]/[t]$, which is the diffusion of the ions of potassium chloride (KCl). In the simulations done by Schoell the diffusion of the monomers was set to be the same as the one for the ions, to compare our results we do the same. The mobility of the monomers is set indirectly through the friction coefficient of the *Lattice-Boltzmann* fluid γ. The *Einstein-Smoluchoswski* relation gives us

$$D = k_{\mathrm{B}} T \mu \tag{6.13}$$

where D is the diffusion coefficient and $\mu = \frac{1}{\gamma}$ the mobility. With the diffusion coefficient of the ions being $6.515 \cdot 10^{-4}\,[x^2]/[t]$ we get $\gamma = 1535\,[E][t]/[x^2]$, if we want the diffusion of the monomers to match. The simulation is not stable for such a huge friction, therefore we follow Schoell and scale both diffusion and friction by a factor of 100 yielding $D =$

$6.515 \cdot 10^{-2}\,[x^2]/[t]$ and $\gamma = 15.35\,[E][t]/[x^2]$ to get a stable simulation. All parameters are also listed in table 6.2 and the unit system is defined in table 6.1. After an initial equilibration of 10^6 time steps we let the simulation run for 10^7 time steps and measure the observables every 1000 steps. These are the same simulations times that proved to be reasonable in Schoells simulations. Fig. 6.3 shows the End-to-End distance R_e, radius

Table 6.1: Unit system

	Simulation unit	SI unit
mass $[m]$	$3.04 \cdot 10^{-25}$ kg	1 kg
length $[x]$	2.5 Å	1 m
time $[t]$	$2.14 \cdot 10^{-14}$ s	1 s
energy $[E]$	$4.14 \cdot 10^{-21}$ J	1 J
charge [q]	$1.602 \cdot 10^{-19}$ C = 1 e	1 C

Table 6.2: Simulation Parameters

Parameter	in simulation units	in SI unit
time step	$0.01[t]$	$2.14 \cdot 10^{-15}\,\mathrm{s}$
box size	$40 - 98[x]$	$10^{-8} - 2.45 \cdot 10^{-8}\,\mathrm{m}$
Number of Monomers	3-50	
monomer concentration	$4.7 \cdot 10^{-5} 1/[x^3]$	5 mM
grid size	$1[x]$	2.5 Å
ion valency	$\pm 1[q]$	$1.602 \cdot 10^{-19}\,\mathrm{C}$
l_B	$2.84[x]$	7.11 Å
D_{ions}	$6.515 \cdot 10^{-2}\,[x^2]/[t]$	$1.9 \cdot 10^{-5}\mathrm{cm}^2/\mathrm{s}$
σ (WCA)	$1[x]$	2.5 Å
ϵ (WCA)	$0.25[E]$	0.625 Å
Δr (FENE)	$1.5[x]$	3.74 Å
K (FENE)	$30[E]/[x^2]$	$1.99\,\mathrm{J}/\mathrm{m}^2$

of gyration R_g and the hydrodynamic radius R_h plotted over the number of bonds $N - 1$. The estimated statistical errors are very small ($< 0.1\%$) and thus not visible at the scale of the figures. The added noise term does not show any relevant impact on the results.

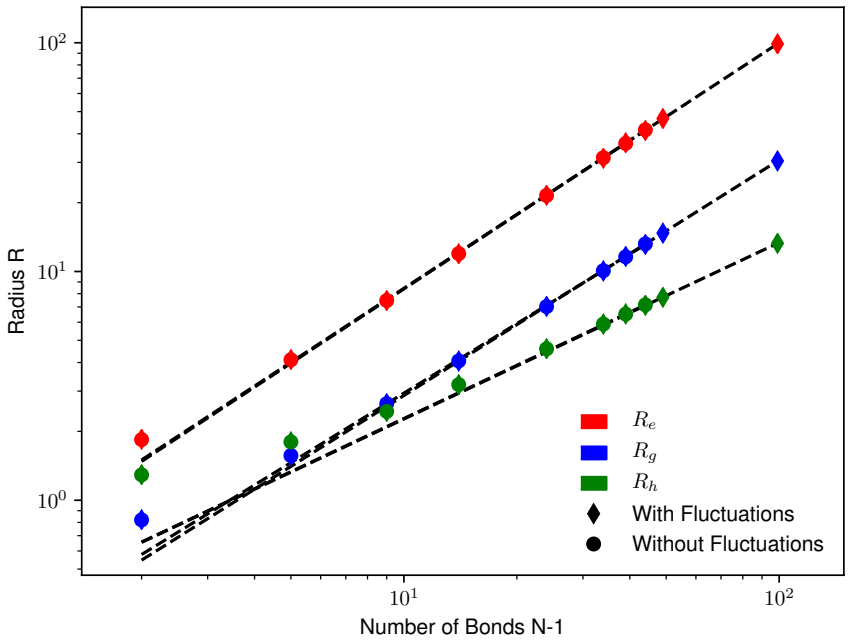

Figure 6.3: Comparison of the Scaling exponents of the end-to-end distance R_e (red), radius of gyration R_g (blue) and hydrodynamic radius R_h (green) for simulations with (diamonds) and without (circles) thermal fluctuations in the ion fluxes.

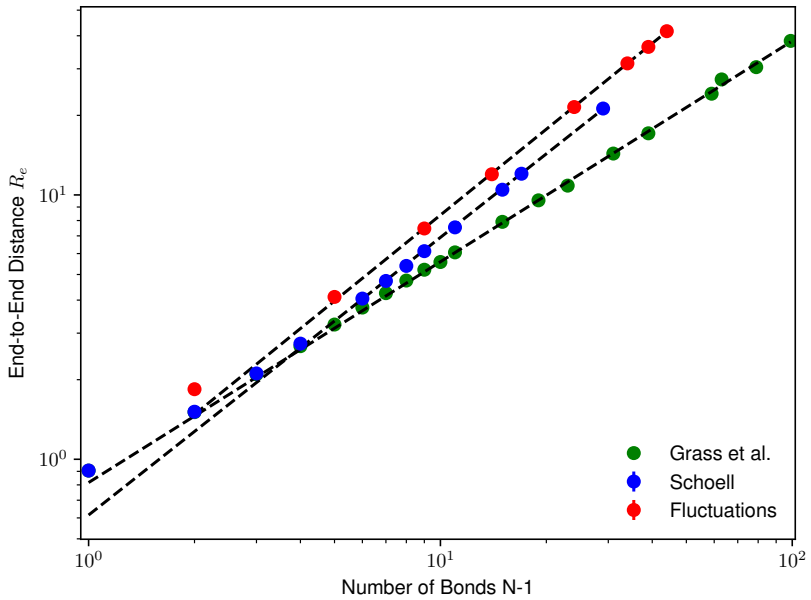

Figure 6.4: Comparison of the Scaling exponents of the end-to-end-distance R_g for the continuum based simulations without thermalised ion fluxes done by Schoell [27] (blue) and this author (red) and the particle based simulations done by Grass [13] and Grass et. al. [12] (green).

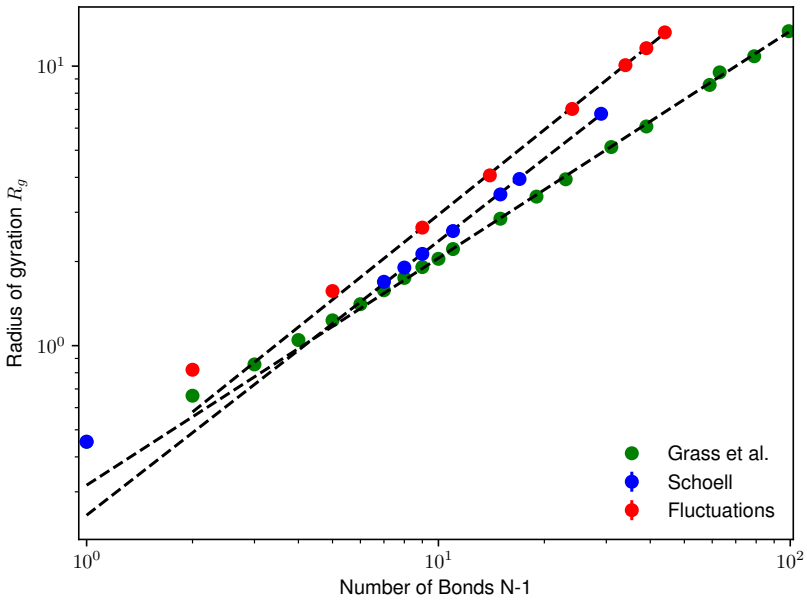

Figure 6.5: Comparison of the Scaling exponents of the radius of gyration R_g for the continuum based simulations without thermalised ion fluxes done by Schoell [27] (blue) and this author (red) and the particle based simulations done by Grass [13] and Grass et. al. [12] (green).

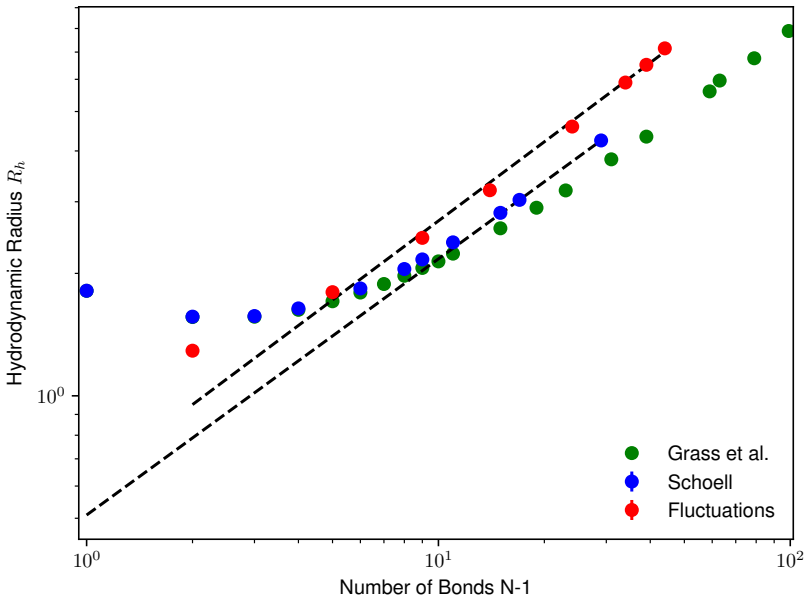

Figure 6.6: Comparison of the Scaling exponents of the hydrodynamic radius R_g for the continuum based simulations without thermalised ion fluxes done by Schoell [27] (blue) and this author (red) and the particle based simulations done by Grass [13] and Grass et. al. [12] (green).

We also compare our results to Schoell [27] and the particle based simulations of Grass et al. [13] [12] (see Figures 6.4-6.6). The in Tab. 6.3 listed scaling factors for the End-to-End distance and the radius of gyration are still ≈ 1, which represents a very linear conformation we would only expect for a system with no counterions at all. The hydrodynamic radius does not seem to follow the scaling behaviour for the simulated polymer lengths.
The counterions present should shield some of the electrostatic interaction between the monomers giving us smaller scaling coefficients but the results do not reflects this. Instead of further polymer simulations we focus on analysing possible problems with the model that could lead to the shown results.

Table 6.3: Scaling coefficients, with(out) fluc. refers specially to the fluctuations of the ionic fluxes

			This thesis	
Scaling coefficient of	Schoell [27]	Grass et al. [13][12]	With fluc.	Without fluc.
R_e	1.05	0.84	1.08	1.08
R_g	0.98	0.81	1.03	1.03
R_h	0.63	-	0.77	0.77

6.3 Problem analysis

Through the analysis of the polymer simulations a problem with the electrostatic interaction between the MD particles and the ionic species of the EK model was found to be a possible source for the incorrect results. Specifically the mapping of the point charges to the discrete EK model turns out to be highly problematic for systems in which the distances between point charges are in the order of only one or two lattice lengths. The current implementation maps the charge density of the particle to surrounding EK cells while conserving the centre of mass (see Figure 6.2).

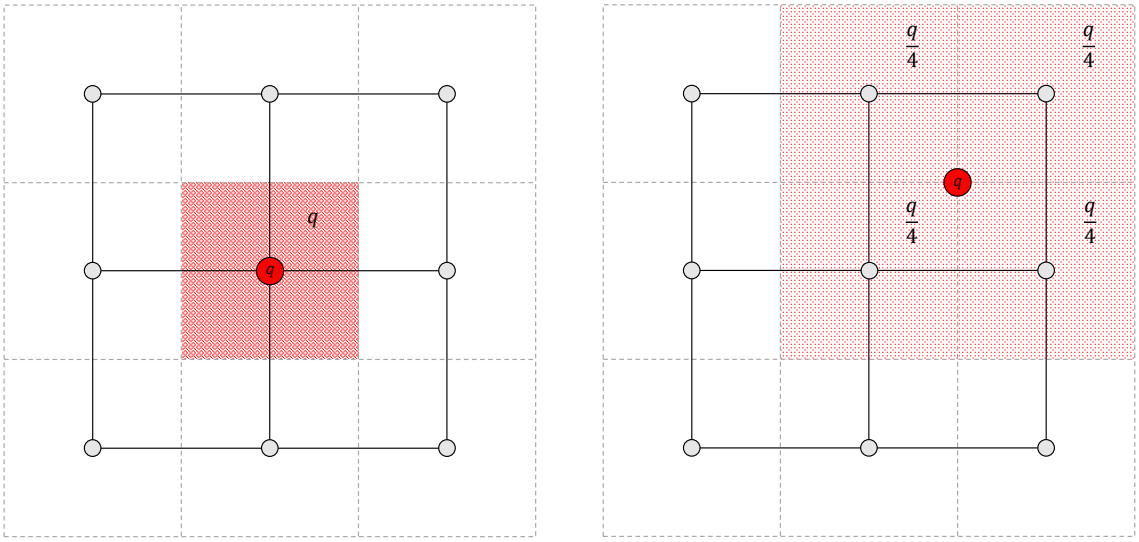

Figure 6.7: Mapping of a point charge (Red circle) to the cells of the EK model. The dashed lines indicate the borders of the cells associated with the lattice nodes (grey circles). Left: The point charge sits exactly on a lattice node. Right: The point charge sits in between cells, which means the charge will be distributed between the four surrounding cells.

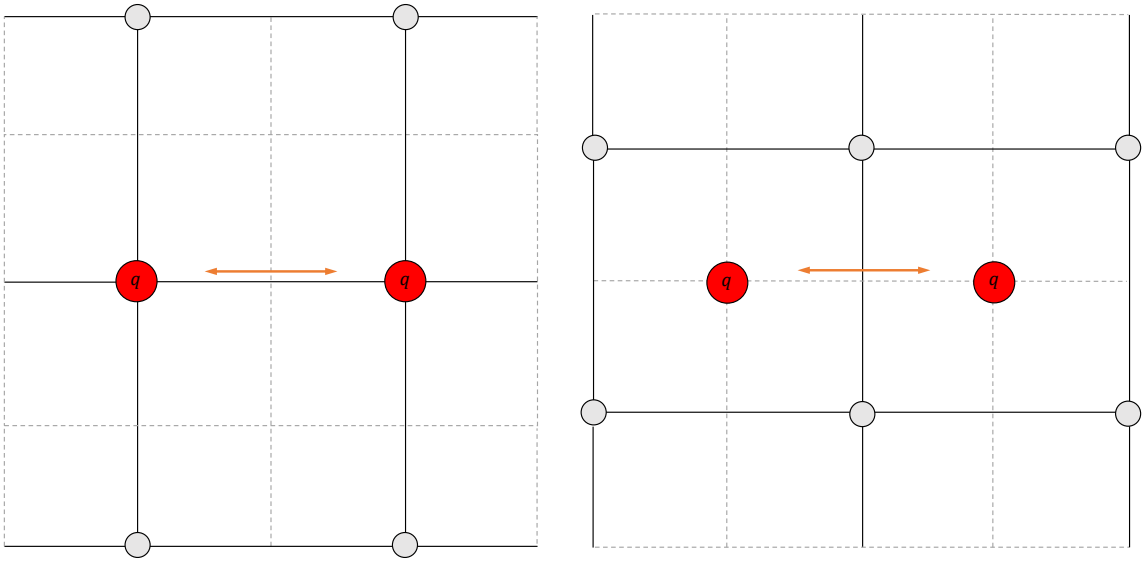

Figure 6.8: On the left side the point charges sit exactly on a lattice node and on the rights side the point charges sit in between cells. The dashed lines indicate the cells with the lattice nodes in their center (grey circles)

Figure 6.7 shows the mapping in two dimensions for a point charge sitting in the exact center of a cell and for a point charge sitting on a corner. In our polymer simulations we expected the counterions to accumulate around the charged monomers, shielding the monomers from the electrostatic interaction with each other. The simulations showed nearly completely stretched polymers, because the incorrect electrostatic forces led to a smaller accumulation of counterions and therefore a stronger repulsion between monomers.

In order to investigate this problem we look at a simple system consisting of two particles with charge q and an electrokinetic species with charge $-q$ neutralizing the system. The particles are fixed in space, any interactions with the background fluid and the fluctuations of the counterions are switched off. We only consider the electrostatic interactions, for the interaction between particles we again use the P3M method. We measure the force acting on the particles when both are placed in the centre of a cell and when both are placed on the corner of a cell (see Fig. 6.8).
In the following Figures a negative force is defined to be a force pointing towards the other particle, so it is an attractive force between the particles and a positive force means the two particles repel each other.

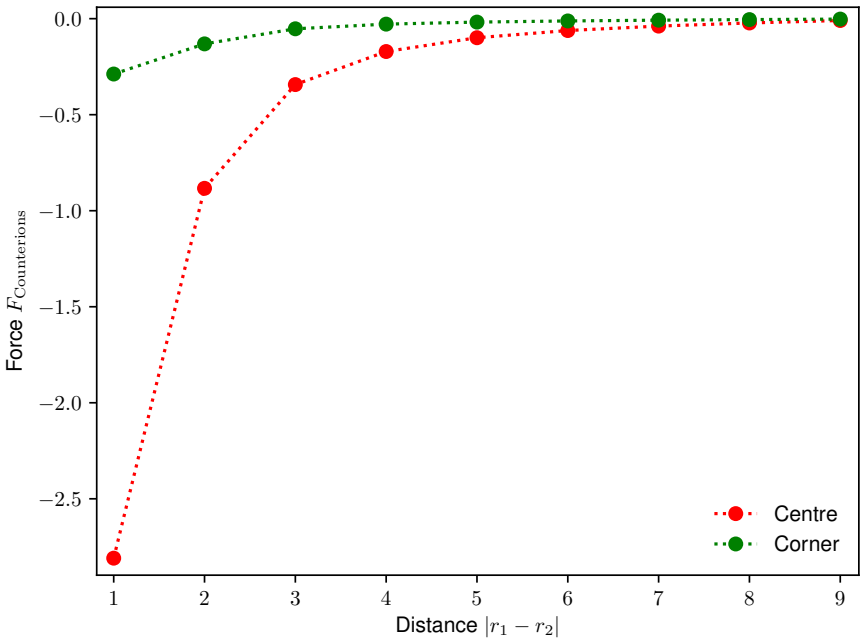

Figure 6.9: Force on the particles caused by the counterions. A negative force points to the other particle, meaning the particles attract each other. For the center case more counterions accumulate around each of the particles resulting in a stronger attractive force.

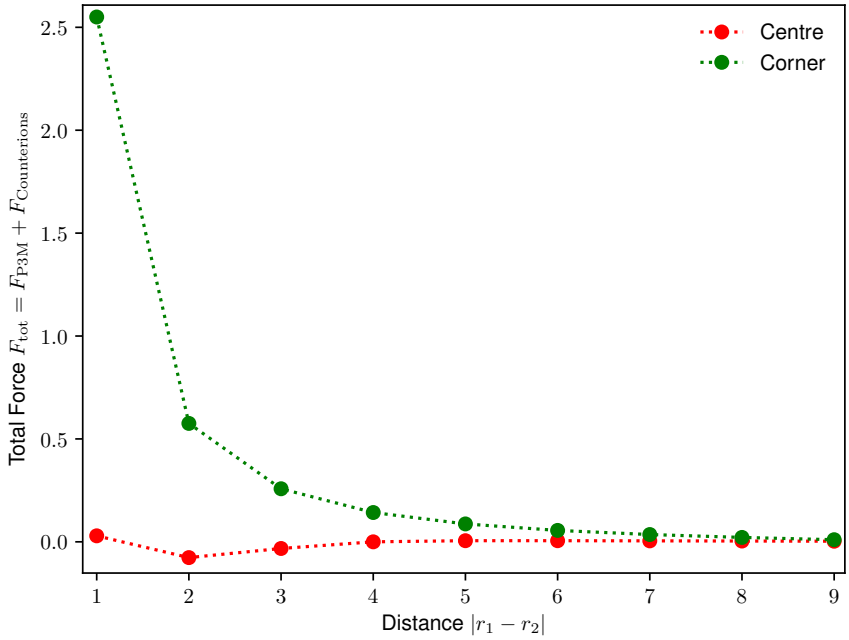

Figure 6.10: Total force on one of the particles resulting from the electrostatic interaction. A positive force points away from the other particle, meaning the particles repel each other. For the corner case the counterions do not properly shield the electrostatic repulsion between the two particles resulting in a net repulsive force.

Figure 6.9 shows the force on the particles resulting only from the counterions. There is a very noticeable difference between the two cases. The counterions accumulating around both particles result in an attractive force between the particles. The force is several times larger for the center case showcasing that more counterions accumulate around the particles if placed in the center of a cell.
Figure 6.10 shows the total force on the particles caused by the particle-particle interaction and the interaction with the counterions. We again see a big difference between the two cases. For the center case the attractive force caused by the counterions and the repulsive force caused by the electrostatic interaction of the two particles with each other compensate each other. This results in a small total force even for small distances. For the corner case we get a strong repulsive force since the attractive force is a lot smaller as shown in Fig. 6.9. Only for a distance of nine cells between the particles the force approaches the same value. For the polymer simulations described in the previous section the typical distance between monomers was about a single cell size making this problem very relevant.
We repeat the two particle simulation with better resolutions. An increase in resolution should be able to give better results, but as Fig. 6.11 shows we would need to use an eight

times higher resolution to reach an acceptable error. Considering the time such a simulation would take with the current implementation of the EK model this would make the model slower than a simulation using particle ions even for very high salt concentrations. Further investigations and improvements are needed.

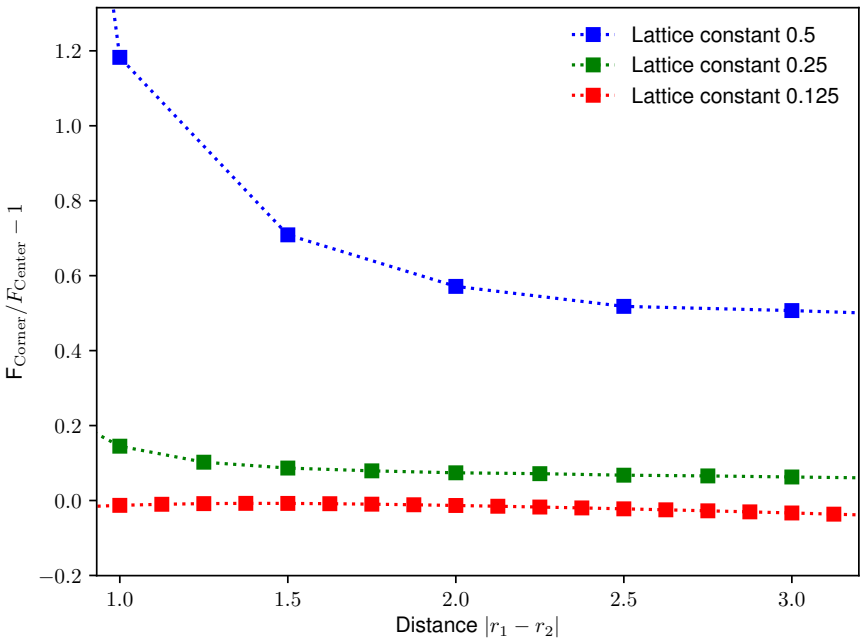

Figure 6.11: Relative difference of total force on a particle for the corner and the center case for different resolutions.

Chapter 7

Conclusion and Outlook

Not explicitly simulating every ion in a system and instead including them through a continuum model provides a significant advantage for the simulation of huge systems. A continuum model such as the one discussed in this thesis can provide help in understanding and predicting the dynamics and properties of polymers like DNA or proteins and other systems where all atom simulations are very limited by computational power.
We investigated the thermalisation of a continuum model for ion species first introduced by Capuani et. al. [4]. A noise term derived from independent Brownian motion of the particles was introduced. We implemented the fluctuations into the existing electrokinetic solver of the ESPResSo simulation package. We utilize CUDA capable GPUs for the massive computational power these devices represent. A new discretization was also introduced and implemented.
We showed that the validity of the discussed thermalisation is limited for small densities, which is expected since the Poisson distribution our fluctuations reproduce no longer resembles a continuous function in such systems. The chosen statistical mechanic approach is poor in situations in which we have less than one particle per cell.
We verify the thermalisation by investigating two test systems for which we either provide an analytical solution or compare results with a MD simulation also done using ESPResSo. The simplest system is an ideal gas for which we expect the density to have a binomial distribution or in certain limits a Poisson distribution. The other system is a gas consisting of two oppositely charged species, with no other forces or potentials outside of the electrostatic interaction. For both system we were able to show good agreement with either an analytical solution or a MD simulation.
The discretization was investigated by simulating a point source like system and comparing it with the analytical solution and the previously implemented stencil. The results again match very well.
We simulated a model of coarse-grained polyelectrolytes and analysed the scaling coef-

ficients for end-to-end distance, radius of gyration and the hydrodynamic radius. Such simulations were already done by Schoell [27] using the previous implementation of the model, but the obtained values for the scaling coefficients were not matching the correct physical behaviour. A repeat of the simulation with the now thermalised counterions yielded the same results showing no significant impact of the fluctuations. The actual source of these errors was tracked down to be a problem concerning the electrostatic interaction of the monomers (MD Particles) and the counterions, which are handled by the in this thesis discussed continuum model. The problem emerges from the mapping of point charges to the lattice of the electrokinetics solver. Depending on the position of a point charge in relation to the lattice the charge will be distributed between different numbers of nodes leading to significant errors for short range interactions. The problem can in theory be compensated by increasing the resolution and making sure that monomers are several lattice lengths apart, but computational time becomes a big concern.
An adaptive refinement method allowing us to only increase the resolution for the relevant area containing the polymer could make simulations times tolerable again. Recently Tischler [28] worked on adaptive Electrokinetics running on CPUs and in the future an implementation utilizing the computational power of GPUs is possible. The implementation of the fluctuations is also not fully optimized yet giving another starting point to reduce simulation times. To summarize, the proposed *Electrokinetics* model with the added noise term in the ion fluxes is now fully thermalised, but further investigations and optimizations regarding the limitations for low densities and the coupling with *Molecular Dynamic* particles are needed.

Bibliography

[1] Aleksei Aksimentiev. "Deciphering ionic current signatures of DNA transport through a nanopore". In: *Nanoscale* 2.4 (2010), p. 468. DOI: 10.1039/b9nr00275h.

[2] Aleksij Aksimentiev et al. "Microscopic Kinetics of DNA Translocation through Synthetic Nanopores". In: *Biophysical Journal* 87.3 (Sept. 2004), pp. 2086–2097. DOI: 10.1529/biophysj.104.042960.

[3] Axel Arnold et al. "Espresso 3.1: Molecular dynamics software for coarse-grained models". In: *Meshfree methods for partial differential equations VI*. Springer, 2013, pp. 1–23.

[4] Fabrizio Capuani, Ignacio Pagonabarraga, and Daan Frenkel. "Discrete solution of the electrokinetic equations". In: *The Journal of Chemical Physics* 121.2 (July 2004), pp. 973–986. DOI: 10.1063/1.1760739.

[5] David S Dean. "Langevin equation for the density of a system of interacting Langevin processes". In: *Journal of Physics A: Mathematical and General* 29.24 (Dec. 1996), pp. L613–L617. DOI: 10.1088/0305-4470/29/24/001.

[6] Markus Deserno and Christian Holm. "How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines". In: *The Journal of Chemical Physics* 109.18 (Nov. 1998), pp. 7678–7693. DOI: 10.1063/1.477414.

[7] Markus Deserno and Christian Holm. "How to mesh up Ewald sums. II. An accurate error estimate for the particle–particle–particle-mesh algorithm". In: *The Journal of Chemical Physics* 109.18 (Nov. 1998), pp. 7694–7701. DOI: 10.1063/1.477415.

[8] Christopher M. Dobson. "Protein folding and misfolding". In: *Nature* 426.6968 (Dec. 2003), pp. 884–890. DOI: 10.1038/nature02261.

[9] Aleksandar Donev et al. "On the accuracy of finite-volume schemes for fluctuating hydrodynamics". In: *Communications in Applied Mathematics and Computational Science* 5.2 (June 2010), pp. 149–197. DOI: 10.2140/camcos.2010.5.149.

[10] Burkhard Dünweg and Anthony J. C. Ladd. "Lattice Boltzmann Simulations of Soft Matter Systems". In: *Advanced Computer Simulation Approaches for Soft Matter Sciences III*. Springer Berlin Heidelberg, pp. 89–166. DOI: 10.1007/978-3-540-87706-6_2.

[11] *ESPResSo Documentation*. 2018. URL: http://espressomd.org/html/doc4.0/index.html.

[12] Kai Christian Grass. "Towards realistic modelling of free-solution electrophoresis: a case study on charged macromolecules". PhD thesis. PhD thesis, Goethe-Universität Frankfurt, 2009.

[13] Kai Grass et al. "Importance of Hydrodynamic Shielding for the Dynamic Behavior of Short Polyelectrolyte Chains". In: *Physical Review Letters* 100.9 (Mar. 2008). DOI: 10.1103/physrevlett.100.096104.

[14] M. Karplus and J. Kuriyan. "Molecular dynamics and protein function". In: *Proceedings of the National Academy of Sciences* 102.19 (May 2005), pp. 6679–6685. DOI: 10.1073/pnas.0408930102.

[15] Changho Kim et al. "Stochastic simulation of reaction-diffusion systems: A fluctuating-hydrodynamics approach". In: *The Journal of Chemical Physics* 146.12 (Mar. 2017), p. 124110. DOI: 10.1063/1.4978775.

[16] Timm Krüger et al. *The Lattice Boltzmann Method: Principles and Practice (Graduate Texts in Physics)*. Springer, 2016. ISBN: 978-3-319-44649-3.

[17] Michael Kuron et al. "Moving charged particles in lattice Boltzmann-based electrokinetics". In: *The Journal of Chemical Physics* 145.21 (Dec. 2016), p. 214102. DOI: 10.1063/1.4968596.

[18] Yves Lansac, Prabal K. Maiti, and Matthew A. Glaser. "Coarse-grained simulation of polymer translocation through an artificial nanopore". In: *Polymer* 45.9 (Apr. 2004), pp. 3099–3110. DOI: 10.1016/j.polymer.2004.02.040.

[19] Hans-Jörg Limbach et al. "ESPResSo—an extensible simulation package for research on soft matter systems". In: *Computer Physics Communications* 174.9 (2006), pp. 704–727.

[20] Saman Monjezi et al. "Computational Studies of DNA Separations in Micro-Fabricated Devices: Review of General Approaches and Recent Applications". In: *Advances in Chemical Engineering and Science* 07.04 (2017), pp. 362–392. DOI: 10.4236/aces.2017.74027.

[21] Nvidia. *CUDA Toolkit Documentation*. 2018. URL: https://docs.nvidia.com/cuda/.

[22] Randall C O'Reilly and Jeffrey M Beck. "A family of large-stencil discrete Laplacian approximations in three dimensions". In: *Int. J. Numer. Methods Eng* (2006), pp. 1–16.

[23] Bernt Øksendal. *Stochastic Differential Equations*. Springer Berlin Heidelberg, 2003. DOI: 10.1007/978-3-642-14394-6.

[24] Georg Rempfer. "A Lattice based Model for Electrokinetics". Master's thesis. University of Stuttgart, 2013.

[25] Georg Rempfer et al. "Reducing spurious flow in simulations of electrokinetic phenomena". In: *The Journal of Chemical Physics* 145.4 (July 2016), p. 044901. DOI: 10.1063/1.4958950.

[26] D. Roehm and A. Arnold. "Lattice Boltzmann simulations on GPUs with ESPResSo". In: *The European Physical Journal Special Topics* 210.1 (Aug. 2012), pp. 89–100. DOI: 10.1140/epjst/e2012-01639-6.

[27] Simon Schoell. "A Hybrid Particle and Continuum based Simulation Method for Electrokinetics". Bachelor thesis. University of Stuttgart, 2016.

[28] Ingo Tischler. "Implementing adaptive Electrokinetics in ESPResSO". Master thesis. University of Stuttgart, 2018.

[29] Alexander J. Wagner and Kyle Strand. "Fluctuating lattice Boltzmann method for the diffusion equation". In: *Physical Review E* 94.3 (Sept. 2016). DOI: 10.1103/physreve.94.033302.

[30] Florian Weik, Stefan Kesselheim, and Christian Holm. "A coarse-grained DNA model for the prediction of current signals in DNA translocation experiments". In: *The Journal of Chemical Physics* 145.19 (Nov. 2016), p. 194106. DOI: 10.1063/1.4967458.

Acknowledgements

I want to thank

- Christian Holm, head of the ICP, for the opportunity to work on this thesis.
- Georg Rempfer and Florian Weik for their supervision and help during the past year.
- All colleagues at the ICP who created a pleasant environment to work in.
- My friends and family for their support during all my years of studying.