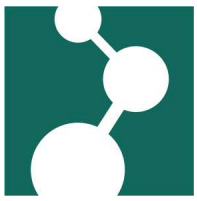
Max–Planck–Institut für
Polymerforschung



Computer simulations of charged systems in partially periodic geometries

Dissertation
zur Erlangung des Grades
"Doktor der Naturwissenschaften"

am Fachbereich Physik
der Johannes Gutenberg–Universität
in Mainz

Axel Arnold

geb. in Ulm

Mainz, im August 2004

Datum der mündlichen Prüfung: 13. Dezember 2004

Zusammenfassung

Diese Arbeit stellt Algorithmen zur Berechnung der elektrostatischen Wechselwirkung in partiell periodischen System vor. Als Rahmen für diese Verfahren dient das Simulationsprogramm ESPResSo, an dessen Entwicklung der Autor maßgeblich beteiligt war. Die wesentlichen Merkmale des Programms werden aufgezählt und der innere Aufbau des Programms erläutert.

Im Anschluss werden Algorithmen für die Berechnung der Coulomb–Summe in dreidimensional periodischen Systemen besprochen. Diese Methoden bilden die Basis für die im folgenden beschriebenen Verfahren für partiell periodische Systeme.

Für Systeme mit einer nichtperiodischen Koordinate wird, ausgehend von der MMM2D Methode, die ELC Methode entwickelt. Diese erlaubt es, mit Hilfe eines Korrekturterms Methoden für dreidimensional periodische Systeme auch bei nur zwei periodischen Koordinaten zu verwenden. Dabei ist die für die Korrektur benötigte Rechenzeit für große Teilchenzahlen vernachlässigbar. Die Leistungsfähigkeit von MMM2D und ELC wird anhand der Implementierungen in ESPResSo demonstriert. Es wird erläutert, wie sich verschiedene dielektrische Konstanten innerhalb und ausserhalb der Simulationsbox realisieren lassen.

Schließlich wird die MMM1D Methode für Systeme mit einer periodischen Koordinate entwickelt. Diese Methode wird auf das Problem der Anziehung gleichnamig geladener Stäbe in der Anwesenheit von Gegenionen angewandt und Ergebnisse der Strong–Coupling–Theorie für die Gleichgewichtsdistanz der Stäbe bei unendlicher Gegenionen–Kopplung mit Hilfe von Computersimulationen überprüft. Der Grad der Übereinstimmung zwischen Simulation bei endlicher Kopplung und Theorie kann durch einen Parameter γ_{RB} charakterisiert werden.

Im Spezialfall $T = 0$ finden sich unter gewissen Umständen flache Konfigurationen, in denen alle Gegenionen in der Stab–Stab–Ebene liegen. Von diesen Konfigurationen wird analytisch die energetisch günstigste und deren Stabilität bestimmt, was von nur einem Parameter γ_z ähnlich γ_{RB} abhängt. Diese Ergebnisse können durch Computersimulationen bestätigt werden.

Summary

This work presents algorithms for the calculation of the electrostatic interaction in partially periodic systems. The framework for these algorithms is provided by the simulation package ESPResSo, of which the author was one of the main developers. The prominent features of the program are listed and the internal structure is described.

In the following, algorithms for the calculation of the Coulomb sum in three dimensionally periodic systems are described. These methods are the foundations for the algorithms for partially periodic systems presented in this work.

Starting from the MMM2D method for systems with one non–periodic coordinate, the ELC method for these systems is developed. This method consists of a correction term which allows to use methods for three dimensional periodicity also for the case of two periodic coordinates. The computation time of this correction term is negligible for large numbers of particles. The performance of MMM2D and ELC are demonstrated by results from the implementations contained in ESPResSo. It is also discussed, how different dielectric constants inside and outside of the simulation box can be realized.

For systems with one periodic coordinate, the MMM1D method is derived from the MMM2D method. This method is applied to the problem of the attraction of like–charged rods in the presence of counterions, and results of the strong coupling theory for the equilibrium distance of the rods at infinite counterion–coupling are checked against results from computer simulations. The degree of agreement between the simulations at finite coupling and the theory can be characterised by a single parameter γ_{RB}.

In the special case of $T = 0$, one finds under certain circumstances flat configurations, in which all charges are located in the rod–rod plane. The energetically optimal configuration and its stability are determined analytically, which depends on only one parameter γ_z, similar to γ_{RB}. These findings are in good agreement with results from computer simulations.

Contents

Introduction

The computer industry is the fastest growing industry nowadays. Similarly, computational physics is one of the youngest fields and fastest growing branches of physics. The first electronic computers were build during the second world war to perform heavy computations involved in the development of nuclear weapons and code breaking. In the early 1950's, the first computers became partially available for civilian use, and one of the first applications were computer simulations. The MANIAC, built by Nick Metropolis for the Los Alamos National Laboratory in the USA, was one of the first of these computers. The laboratory was interested in finding as many applications of their machine as possible, and one of these was the Metropolis Monte–Carlo algorithm for problems in statistical mechanics. Today the range of applications for computers has grown beyond any bounds, and even a standard home PC is many orders of magnitude faster than the MANIAC. According to Moore's law the number of transistors in a processor doubles every four years, which has proven to be true for now four decades, and similarly the speed of modern computers continues to grow.

But what is the physical use of computer simulations? Only a very small portion of the problems arising in physics can be solved analytically. Even the equations of motion for three bodies in simple Newtonian mechanics are essentially unsolvable. The properties of everyday's materials are often determined by the complex interplay of thousands of atoms. Solving the equations of motion exactly for a such a complex system using only pencil and paper is beyond hope, and one has to find different ways to predict the properties of these materials.

Before computer simulations were known, there was only one way to investigate such problems, namely deriving a good theory for an approximative description of the material. However, only very few systems can be treated exactly, such as the ideal gas, the harmonic oscillator and a number of lattice models, e. g. the Ising model. Therefore most properties of real materials were predicted by approximative theories like the Debye–Hückel theory or the Poisson–Boltzmann equation for electrolytes. Given sufficient information on the intermolecular interactions, these theories often give good estimates for the relevant observables and insight into the underlying physical mechanism, but their range of applicability is limited. For example, in chapter 7 a problem will be presented, where the Poisson–Boltzmann theory fundamentally fails.

If theory and experiment disagree, two sources of error are possible: either the model system does not represent the experiment correctly, or the theory does not describe the model system correctly (or both, in the worst case). Here computer simulations can be extremely helpful by providing essentially exact results for a model system, provided the simulation was done properly. If the simulation results agree with the experiment, but not the theory, the theory does not even describe the model system correctly. If the simulation cannot reproduce the experiment, the model system misses

some fundamental features of the experiment. In this sense a computer simulation is an experiment on a theoretical model and provides a link between theory and experiment. This explains to some extent the speed at which the field of computer simulations grows in today's physics.

Another reason for the importance of computer simulations is that some model systems can describe the reality well enough to replace experiments. Three popular examples are the computer aided design ("CAD") of pharmaceuticals, applications of CAD in the automotive industry and weather forecasts. Nowadays pharmaceuticals can be designed to fit, for example, a specific receptor. To this aim basically a large pool of possible candidate substances has to be tested for the binding characteristics to the binding site in mind. Real experiments would take weeks, while powerful computers can give a good estimate of the binding rate within hours. Besides this, computer simulations are often cheaper than real experiments. The latter reasoning is especially important in the car industries. Finite element methods allow for a new car design to go through several hundred simulated crash tests before the first prototype is built, and most parts of the engine are optimised using computer simulations. Today the weather forecast is usually reliable for at least one day, although the underlying physical equations are extremely complicated and the system to simulate is incredibly large.

These examples also show the large range of length scales on which computer simulations are used today. From the submolecular level to the macroscopic level of whole cars or even the global climate. And finally new theories in astronomy can only be tested using computer simulations. Interestingly, the latter type of computer simulations is often more similar to calculations on the microscopic level than for example to finite element methods.

This work is dedicated to simulations of this type, where particles are essentially sphere–like and the interactions are long ranged. The length scales we are interested in range between 1nm and 10μm, which are the relevant length scales for a class of materials collectively called "soft matter".

Soft matter is a term for materials in states that are neither simple liquids nor hard solids of the type studied, for example, in solid state physics. Examples are polymers, colloids, liquid crystals, glasses, and dipolar fluids. Familiar examples of such materials are glues, paints, soaps or baby diapers. Others are important in industrial processes, such as polymer melts that are molded and extruded to form plastics [46]. Most biological materials are soft matter as well — DNA, membranes, filaments and other proteins belong to this class.

All these materials have in common that a wide range of length and time scales is important for their microscopic behaviour as well as their macroscopic properties. Typical energies between different structures are comparable to thermal energies. Hence, Brownian motion or thermal fluctuations play a prominent role. Another key feature of soft matter systems is their propensity to self-assemble. This often results in complex phase behaviours yielding a rich variety of accessible structures which form spontaneously. Most of the biological systems are usually not even in thermal equilibrium but evolve among switchable steady states.

Many properties of soft matter emerge on the molecular rather than the atomistic level. The elasticity of rubber is the result of entropy of the long polymers molecules, and the hygroscopic materials used in modern diapers store water inside a

polyelectrolyte network. In both examples the important properties — elasticity rsp. hygroscopy — are not the result of a chemical process, but rather of physical interplay of the involved molecules. To reproduce this physical interplay in a computer simulation on the atomistic level, one would have to incorporate several millions of atoms, which is only possible on very small time scales even with the most powerful modern computers. But since the properties of these materials do not emerge on the atomistic level, a simpler description of the material is often sufficient. Polymers such as polyelectrolytes or rubber often can be modelled by bead–spring models, i. e. (charged) spheres connected by springs. Each of the spheres stands for several atoms, often a complete monomer or even larger compounds. Although this model hides most of the chemical properties, it is quite successful in the description of polymers. Nevertheless, computer simulations on the bead–spring level still incorporate several thousands of spheres and springs and require an efficient simulation software.

For computer simulations of soft matter a large number of program packages exist, for example polyMD [82], BALL [13], OCTA [24], GISMOS [54], GROMACS [97], GROMOS [98], Amber [77], NWChem [91], DL_Poly [89], LAMMPS [79] or NAMD [71]. These simulation packages feature a large set of physical interactions, constraints and other ingredients, however, the computer simulation of a specific problem often requires some features not yet implemented. Examples are newly developed potentials or constraints.

So far in the author's work group this problem was solved by hard coding these features directly into existing simulation codes. This is highly ineffective, as these codes were not well documented and not designed to allow easy modification. Moreover there were several code trees which were not compatible. Often one feature needed for a new project was implemented in one code tree, while another feature was added to another tree. So although both features were already implemented and tested, they had to be implemented again in a new hybrid code.

In 2002, the author's work group decided on using one single code base for all future simulations. Looking at the simulation packages existing so far, we did not find a package that seemed to meet all our expectations. Some of the codes, e. g. GROMACS or NAMD, have a different focus, namely the atomistic level, while we are primarily interested in simulations of bead–spring models. For these models packages like polyMD or LAMMPS are well suited, but our efforts to add new features to these codes soon failed because of the complex and not well documented structure of these programs. Elementary features such as force calculation routines were deeply interwoven with the integrator code for code optimisation reasons, which makes adding for example a new method for the calculation of the electrostatic interaction a demanding task.

To improve on this situation, a new simulation package was started, ESPResSo [8,9]. The main goal of ESPResSo is to present state of the art algorithms for simulations of bead–spring models in a parallel code that is *extensible*. This extensibility expresses mainly in that readability is preferred over code optimisations in ESPResSo, and that for many of the basic tasks, such as force calculations, standardised interfaces exist. The lack of optimised code is compensated by the use of state of the art algorithms. The main features of ESPResSo are also expressed by the acronym: **E**xtensible **S**imulation **P**ackage for **Res**earch on **So**ft matter.

The extensibility of ESPResSo enables all scientists within our work group to con-

tribute their own code for reuse by others. But even more important is that researchers from other work groups and institutions can contribute as well. To this aim ESPResSo is published under the GNU public license, and is available through our web page [27]. A welcome side effect of the unoptimised and comparatively simple C–code is that ESPResSo is easily portable to other hardware platforms. In chapter 1 the basics of an MD simulation using ESPResSo are described.

A special focus of the author's work group is on the research of charged systems. However, in computer simulations the calculation of the electrostatic interaction requires highly complex algorithms. Therefore a special focus of ESPResSo and also this work is on state of the art algorithms for the calculation of the Coulomb potential under various types of periodic boundary conditions. Chapter 2 gives an overview of methods that can be used under periodic boundary conditions in all spatial dimensions, which are widely applied to simulate bulk systems.

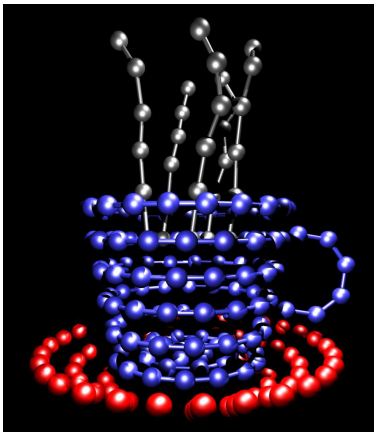
Chapter 3 deals with the case of periodic boundary conditions in only two of the spatial dimensions, as is used for example for studies on membranes or thin films. This case is algorithmically much harder to treat since the spatial symmetry is broken, which is employed at prominent positions in nearly all methods for fully periodic systems, and consequently only a few methods were known so far. Moreover the computationally efficiency of these methods was worse than in the three dimensionally periodic case. In this work a new method for the calculation of electrostatic interaction for such systems, ELC, is developed in chapter 4. Chapter 5 briefly outlines how to deal with different dielectric constants inside and outside of a thin film, which is important for aqueous solutions where one does not treat the water explicitly.

Chapter 6 treats the even more demanding problem of one–dimensionally periodic systems, which can be used in studies of nano–pores or long linear polymers like DNA. Only a few methods for this kind of periodicity are known so far, and their computationally efficiency is even worse than in the two dimensionally periodic case. In that chapter a new method MMM1D is presented, which is derived from the MMM2D method and is fast enough at least for small scale simulations of a few hundred particles.

The last chapter of this work is devoted to an application of MMM1D. It is long known the DNA strands can attract each other although they are highly charged in the presence of multivalent counterions. Mean field theories such as Poisson–Boltzmann fail to predict this attraction, which is generally attributed to counterion correlations. A particularly successful theory in predicting the attractive interaction is the strong coupling theory by A. Moreira and R. Netz. In this work the predictions of the strong coupling theory are compared to computer simulations. These simulations show that the strong coupling theory is valid even for realistic parameters and describes correctly the attraction of two like–charged rods, even though the theory was developed for the limit of infinite coupling. A special case of infinite coupling is the limit of zero temperature. Computer simulations as well as a direct calculation are used to identify the range of validity in the case of zero temperature.

Parts of this thesis have been published in [5, 16] (ELC) and [18, 69] (attraction of like–charged rods).

1 MD-Simulations — ESPResSo



In the second half of 2002, Hanjo Limbach and the author of this work started with the implementation of the simulation core of ESPResSo, half a year later the code was undergoing its first test simulations. The author developed and implemented many of the fundamental parts of the code, such as the particle data organisation, the file I/O or the multiprocessor communication scheme. Especially the particle organisation of ESPResSo has proven to be highly efficient in terms of computational speed.

In the mean time, ESPResSo has been improved dramatically, currently about ten developers work on different parts of ESPResSo, and use the simulation code for research topics as different as biological membranes, polyelectrolytes or polymer networks. The code is used on many different hardware platforms such as PCs (GNU/Linux on IA32 and AMD64 processors), Workstations (MacOS on PowerPC processors and Tru64 on Alpha processors) and high performance servers (AIX on Power4 processors).

The wide field of simulation topics investigated by ESPResSo requires a high flexibility of the simulation code, which in ESPResSo is obtained by a script language which is used to control the simulation process. The simulation control script determines all simulation parameters such as the number and type of particles, the type of interactions between these particles and how the system is propagated. The script language used in these scripts allows to change most of the parameters even during the simulation. This flexibility makes it possible to perform highly complex simulation procedures, such as adapting the interaction parameters to the current configuration during the simulation, cooling down the system in a simulated annealing process, or applying or removing a constraint. An example simulation control script will be discussed in the first section of this chapter.

The following sections deal with the internals of ESPResSo. Sec. 1.3 discusses the particle data organisation scheme of ESPResSo and its advantages. The heart of the simulation is the integrator, which propagates the system in time. Sec. 1.4 describes the velocity Verlet integrator that is used in ESPResSo. The following two sections deal with some of the short and long ranged interactions between particles that are currently implemented in ESPResSo. The long ranged interactions will be discussed in more detail in chapters 2 to 6. Sec. 1.7 finally describes some of the commands that ESPResSo provides for the analysis of the data produced in a simulation.

ESPResSo is not a self contained code, but relies on other open source packages.

Most prominent is the use of the Tcl [93] script language interpreter for the simulation control. For the parallelisation standard MPI routines are used, which on Linux and MacOS are provided e. g. by the LAM/MPI [51] implementation, or MPICH [68]. The P3M method for the electrostatic interaction and the mode analysis tool for membranes rely on the FFTW [30]. Besides this libraries, which are needed to get ESPResSo running at all, the development process is supported heavily by the use of the CVS version control system [14], which allows a large number of developers to work simultaneously on the code.

This chapter gives only an overview of the current features of ESPResSo. The list is by no means complete as there is still a lot of work in progress. This overview focuses on features that are either unique to ESPResSo or will be used in the computer simulations shown in chapters 3 through 7. This chapter is not meant to present all the details of a MD simulation. For this refer e. g. to the textbooks of Allen and Tildesley [3] or of Frenkel and Smit [31].

1.1 Simulation control

A simulation is not only determined by some simple numerical parameters, but by the complete program flow. For example the simulated annealing to zero temperature is done by heating up and cooling down the system several times, and it is often necessary to gradually turn on interactions at the beginning of a simulation to equilibrate the system smoothly. In ESPResSo this problem is solved by means of a script control language. ESPResSo uses Tcl as command language, since Tcl integrates smoothly with C–code and is easy to learn.

To demonstrate the use of the script language, now an ESPResSo control script for the NVT simulation of a Lennard-Jones liquid is given. With a small additional header (a she–bang, "#!") the script can even be called directly from the command line like any other program, which is very convenient for submitting several jobs at once.

In the example script, first some Tcl variables are set which will be used later in the program to determine the simulation parameters (lines starting with a "#" are Tcl comments):

```
# size of the cubic simulation box
set box_length 10.7437
# density of the liquid
set density 0.7
```

The next lines define some parameters of the simulation, the time step, box length and the skin depth. The latter parameter is needed for the link cell algorithm (see Sec.1.3), but has no influence on the physics of the simulated liquid.

```
setmd time_step 0.01
setmd skin      0.4
setmd box_l $box_length $box_length $box_length
```

Now particles are put into the simulation box. The first two lines show how mathematical expressions can be used inside Tcl. Here the number of particles is determined

from the box size and the density. The particles are added particle for particle at random positions in a loop, using the builtin random number generator of ESPResSo. In this example only the properties "position" and "type" are set, other properties are e. g. velocity or charge.

```
set volume [expr $box_l*$box_l*$box_l]
set n_part [expr floor($volume*$density)]

for {set i 0} { $i < $n_part } {incr i} {
    set posx [expr $box_l*[t_random]]
    set posy [expr $box_l*[t_random]]
    set posz [expr $box_l*[t_random]]

    part $i pos $posx $posy $posz type 0
}
```

The Langevin thermostat (see Sec. 1.4) is activated for a temperature of $1kT$ and a friction constant $\Gamma = 1/\tau$ by the command

```
set temp 1
set gamma 1
thermostat langevin $temp $gamma
```

A purely repulsive Lennard-Jones interaction (see Sec. 1.5) between all particles of type 0, which are all in our case, is defined by

```
set lj1_eps     1.0
set lj1_sig     1.0
set lj1_cut     1.12246
set lj1_shift   [expr 0.25*$lj1_eps]
inter 0 0 lennard-jones $lj1_eps $lj1_sig $lj1_cut $lj1_shift 0
```

This interaction is used widely in computer simulations and is a smooth approximation of a hard core interaction between two spheres of diameter 1.

Now a simulation loop for 1 million time steps could look like

```
for {set i 0} { $i < 1000 } { incr i} {
    puts "step $i ftime=[setmd time] energy=[analyze energy total]"
    puts "temp = [expr [analyze energy kinetic]/(1.5*[setmd n_part])]"
    integrate 1000
}
```

Every thousand time steps the simulation time, total energy and the current temperature are printed out. Of course this is not enough for a real simulation, as one normally wants to write out simulation data to configuration files. To this aim ESPResSo has commands to write simulation data to a Tcl stream. The format written by these commands is called *blockfile* in the following. One could add the following lines into the simulation loop to write configuration files "config_0" through "config_999":

```
set f [open "config_$i" "w"]
blockfile $f write variable {time_step skin}
blockfile $f write tclvariable {box_length density}
set temp [expr [analyze energy kinetic]/(1.5*[setmd n_part])]
puts $f "\{energy [analyze energy total] $temp\}"
blockfile $f write particles {id pos type}
close $f
```

The created files "config_..." look like

```
{variable
        {time_step 0.01}
        {skin 0.4}
}
{tclvariable
        {box_length {10.7437}}
        {density {0.7}}
}
{energy 5380.6 1.00005}
{particles {id pos type}
        {0 11.5573 8.87179 4.80079 0}
        {1 7.04209 2.96786 3.74511 0}
        {2 8.08042 4.91621 6.53135 0}
        .
        .
        .
        {864 3.64869 8.02398 3.13255 0}
        {865 7.66632 14.4887 1.31884 0}
        {866 -0.654808 11.9669 0.43845 0}
        {867 6.43409 5.75895 5.46044 0}
}
```

As one can see, the format of a blockfile corresponds to Tcl list of tagged data sets, which are called *blocks*, and allows for adding own data sets like the "energy" block in this case. Depending on the simulation, much more complex observables can be written out here for an easy inspection during the simulation run.

Reading in the blockfiles is automated in ESPResSo. The tagged structure of a blockfile allows an easy identification of the portions ESPResSo knows about and proper parsing. The command

```
set f [open "config_999" "w"]
while { [blockfile $f read auto] != "eof" } {}
close $f
```

will read in the variables time_step and skin and the Tcl variables box_length and density back in, as well as all particles positions and types. The "energy" block is not a standard ESPResSo block and ignored by the read loop.

The simulation as presented above will very likely not run smoothly, since the particles are placed randomly and therefore could overlap. The Lennard–Jones potential is singular for vanishing particle distance, therefore overlapping particles have an extremely high interaction energy. The high initial potential energy will in turn accelerate the particles to velocities larger than what can be treated with the chosen time step, and the simulation might crash. To avoid this, ESPResSo allows to cap the Lennard–Jones interaction, i. e. below a certain distance the potential only grows linearly while the Lennard–Jones force is constant. This force cap is gradually raised in an equilibration loop which might look like

```
set cap 10
while {[analyze mindist] < 0.95} {
    inter ljforcecap $cap
```

```
    integrate 1000
    incr cap 30
}
inter ljforcecap 0
```

This loop will gradually increase the force cap until the minimal distance is larger than 0.95 and then switches off the force cap.

The code example given before is obviously a very simple one. More complicated simulation scripts like the ones used for the data presented in chapter 7 can easily extend over hundreds of code lines. This simulations calculate the parameters for about 10000 simulations by a bisection algorithm implemented in Tcl. Of course this could also have been done using any other simulation package, but the implementation of uncritical parts like the bisection in a script language is much easier compared to a modification of the core simulation program. The necessity to access the average forces on some of the particles would have made the implementation of the bisection in polyMD or LAMMPS a formidable task. The possibility to perform non–standard simulations without the need of modifications to the simulation core was one of the main reasons why we decided to use a script language for controlling the simulation core.

1.2 Internal program flow

The Tcl script is interpreted by an Tcl interpreter on exactly one of the nodes, called the master node or node 0 in the following. All other nodes are called slaves and just wait for commands from the master node. The Tcl interpreter will call C–procedures on the master node, which in ESPResSo by convention have the same name as the corresponding Tcl command, i. e. `setmd` will call the C–procedure "`setmd()`", with the variable name and value as (string) parameters. The C–procedure then parses the input and does something appropriate, in this case setting the value of a global variable. But this is not all — in a multiprocessor environment the change of a parameter has to be communicated to all nodes. Therefore the master node issues an command to the slaves to change the value of the corresponding variable, too. Since the number and order of commands issued by the master node is not known at compile time, the communication during the script execution is asynchronous.

Another important point is that ESPResSo allows changing parameters at any time, even during the simulation run, which might require additional work to be done to ensure consistency. Changing the processor grid for example might change the association of the particles to the different nodes and therefore requires a reorganisation of the internal particle structures. If the Bjerrum length is changed, the currently used method for the calculation of the electrostatic interaction has to be reinitialised. Moreover, if this happens during the simulation, the forces stored in the particles for the current configuration are invalid and have to be recalculated.

Because of the large number of algorithms implemented in ESPResSo, the dependencies are actually much more complex. They are resolved by handler procedures, for example "`on_parameter_change()`", which is called whenever a parameter such as the box length changes, or "`on_coulomb_change()`", which is called every time a parameter of an electrostatic method is changed (inside "`on_parameter_change()`").

To keep the number of these handlers small, they are written in a quite general fashion and often reinitialise more than would be necessary. This small drawback in time is more than compensated by the ease with which changes can be propagated safely through the code and the great flexibility at the script level this allows for.

One of the asynchronous commands starts the propagation of the system in time, the integration. During the integration, however, ESPResSo uses a synchronous communication scheme as any other simulation program for efficiency reasons. Synchronous communication means that every node has to know without prior request which MPI communication follows, i. e. the codes running on all nodes have to be synchronised. This is less robust than the asynchronous communication scheme, but the request–answer structure creates too much overhead and is not needed in the integration when the implementation has been done properly.

1.3 Data organisation — Link cells and Verlet lists

The data ESPResSo uses during the integration are mainly particles, interactions and constraints. While the number of interactions and constraints is normally small and simple lists are efficient enough for their storage, the particle data needs some more elaborate organisation, which will be presented here as it is unique for ESPResSo. A particle itself is represented by a structure consisting of several substructures, which in turn represent basic physical properties such as position, force or charge. The particles are organised in one or more particle lists on each node, called *cells*. The cells in turn are arranged by several possible systems, which are called *cell systems* in the following. A cell system defines a way the particles are stored in ESPResSo, i. e. how they are distributed onto the processor nodes and how they are organised on each of them. ESPResSo currently knows three cell systems, namely an N^2–model, a layered model and the domain decomposition model, which is used in most simulations. The cell models will be discussed in more detail in the following.

Technically, a cell is organised as a dynamically growing array, not as a list. This ensures that the data of all particles in a cell is stored contiguously in the memory. The particle data is accessed transparently through a set of methods common to all cell systems, which allocate the cells, add new particles, retrieve particle information and are responsible for communicating the particle data between the nodes. Therefore most portions of the code can access the particle data safely without direct knowledge of the currently used cell system. Only the force, energy and pressure loops are implemented separately for each cell model, as one can draw significant advantage in the calculation from the particle organisation. For example the computation of the pair forces can be implemented in linear computation time using the domain decomposition method, which will be explained now.

The *domain decomposition cell system* is based on the *link cell algorithm* [37], which will be briefly reviewed now. Many pairwise interactions, such as the Lennard–Jones interaction, are short ranged, i. e. their value is small enough to be neglected at a distance much smaller than the size of the simulated system. Therefore the interactions have to be calculated only with particles close by, but since one has to check the particle distance for every pair of particles, this would still result in a algorithm with a computation time scaling of $\mathcal{O}(N^2)$, i. e. the number of necessary operations

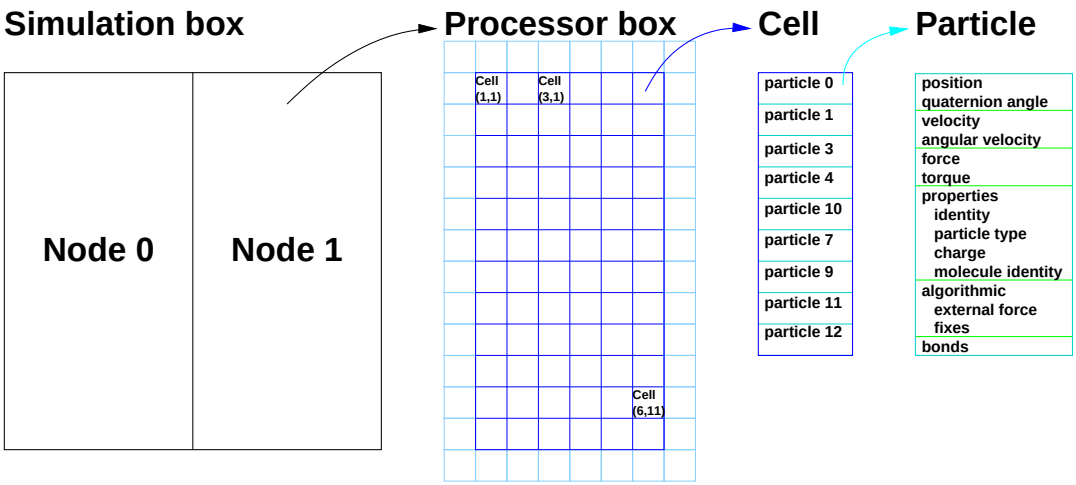


Figure 1.1: Schematic description of the particle data organisation using the domain decomposition cell system within ESPResSo. The simulation box is split up into equally sized regions assigned to a single processor each. These regions are in turn split up into cells (dark blue squares). Around the cells the ghost shell is wrapped, which contains data from the neighbouring processors (light blue squares). The cells are simple arrays of particles, which finally consist of several substructures containing the position, force and other particle data.

grows quadratically with the number of particles. But one can do better: At the beginning of the simulation, so called *link cells* are set up. These cells cover the entire simulation box and contain links to all particles in their spatial domain. Their size is chosen slightly larger than the maximal interaction range. In the force calculation only interactions between particles in adjacent cells have to be calculated, so that one only has to run through the 27 neighbours for each cell (in three dimensions). At constant density and interaction ranges, the average number of particles in a cell is constant and therefore the overall algorithm has order N (since we run once through all cells and therefore all particles).

The force calculation for this system can easily be parallelised as well: First the simulation box is split up equally into as many smaller boxes as processors are available, and each of these boxes is assigned to a processor. The particles are then assigned to the processor that is responsible for the box they are located in. Now these processor boxes are divided up into cells as described above. Each node calculates the interactions for all particles assigned to it. But to do this, the particles located in the cells adjacent to the processor box are needed, too. This shell of cells around a processor box is called the *ghost shell*, and the particles in the ghost shell are called *ghost particles*. The ghost particles have to be communicated between the nodes, while all other particles have to be known only on their respective node. For a schematic representation of the particle organisation used for the domain decomposition, see Fig. 1.1.

After propagating the system by one time step, some particles may have moved out of their cell, which would require to update the cell lists. In a multiprocessor

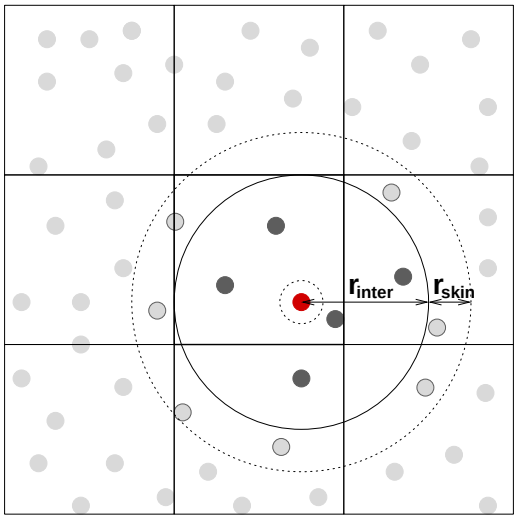


Figure 1.2: Scheme of the link cell and Verlet list algorithm in two dimensions. Interactions with the red particle occur only within the inner, solid circle. However, the Verlet list for the red particle contains all particles inside the dotted circle. The link cell algorithm will consider all particles shown in the figure as candidates for the Verlet list of the red particle, but no other particles.

simulation, that could even mean shifting around particles from processor node to processor node. To reduce the number of particle reorganisations, one can exploit that the cell size is slightly larger than the maximal interaction range. The difference between cell size and maximal interaction range is called the *skin* distance r_{skin}. As long as no particle has moved further than $r_{skin}/2$, the distance of two particles did not increase by more than r_{skin}, so that adjacent cells still contain all interacting particle pairs, and the link cells do not have to be updated. Therefore, with each particle one stores the position where the last sorting process took place, and checks whether this position is further away from the current position than $r_{skin}/2$. If this happens, the link cells are rebuild. The optimal value of r_{skin} is hardware dependent as it reflects the tradeoff between having to treat more particles pairs due to the increased cells size versus having to update the link cells. The skin can be as large as $20 - 40\%$ of the maximal interaction range, and normally the lists are updated about every $20 - 40$ time steps using ESPResSo on a Linux PC. The skin value has to be determined manually by e. g. running several integrations at different skin values, since the optimal value depends heavily on the underlying hardware, especially on the performance of the inter–node communication.

Another trick to further speed up the force calculation are *Verlet lists*. For short range interactions, the possible interaction partners of a single particle are located in a sphere inside the cells surrounding the particle. The size of this sphere is 80% smaller than the size of the full 27 boxes. Therefore it is advantageous to determine all particle pairs within the maximal interaction range plus the skin once, and store a list of these pairs to be used until the next sorting procedure takes place. These particle pair lists are called *Verlet lists*.

The domain decomposition cell system of ESPResSo implements the link cell algorithm. But instead of just having links to the particles in the cells, the cells contain the particles themselves in an array. For an example let us assume that the simulation box has size $20 \times 20 \times 20$ and that we assign 2 processors to the simulation. Then each processor is responsible for the particles inside a $10 \times 20 \times 20$ box. If the maximal interaction range is 1.2, the minimal possible cell size is 1.25 for 8 cells along the first coordinate, allowing for a small skin of 0.05. If one chooses only 6 boxes in the

first coordinate, the skin depth increases to 0.467. In this example we assume that the number of cells in the first coordinate was chosen to be 6 and that the cells are cubic. ESPResSo would then organise the cells on each node in a $6 \times 12 \times 12$ cell grid embedded at the centre of a $8 \times 14 \times 14$ grid. The additional cells around the cells containing the particles represent the ghost shell in which the information of the ghost particles from the neighbouring nodes is stored. Therefore the particle information stored on each node resides in 1568 particle lists of which 864 cells contain particles assigned to the node, the rest contain information of particles from other nodes.

If one particle has moved further than the $r_{skin}/2$, all particles have to be sorted into their correct cells, as discussed above. In the case of ESPResSo, this does not only mean to change the pointers in the cells, but the particle data has to be physically moved to another particle list. This creates obviously considerably more overhead than just changing link pointers.

The advantages of this data organisation become clear in the context of modern computers. Most modern processors have a clock frequency above 1GHz and are able to execute nearly one instruction per clock tick. In contrast to this, the memory runs at a clock speed around 200MHz. Modern double data rate (DDR) RAM transfers up to 3.2GB/s at this clock speed (at each edge of the clock signal 8 bytes are transferred). But in addition to the data transfer speed, DDR RAM has some latency for fetching the data, which can be up to 50ns in the worst case. Memory is organised internally in pages or rows of typically 8KB size. The full 2×200 MHz data rate can only be achieved if the access is within the same memory page (page hit), otherwise some latency has to be added (page miss). The actual latency depends on some other aspects of the memory organisation which will not be discussed here, but the penalty is at least 10ns, resulting in an effective memory transfer rate of only 800MB/s. To remedy this, modern processors have a small amount of low latency memory directly attached to the processor, the *cache*.

The processor cache is organised in different levels. The level 1 (L1) cache is built directly into the processor core, has no latency and delivers the data immediately on demand, but has only a small size of around 128KB. This is important since modern processors can issue several simple operations such as additions simultaneously. The L2 cache is larger, typically around 1MB, but is located outside the processor core and delivers data at the processor clock rate or some fraction of it.

In a typical implementation of the link cell scheme the order of the particles is fairly random, determined e. g. by the order in which the particles are set up or have been communicated across the processor boundaries. The force loop therefore accesses the particle array in arbitrary order, resulting in a lot of unfavourable page misses. In the memory organisation of ESPResSo, the particles are accessed in a virtually linear order. Because the force calculation goes through the cells in a linear fashion, all accesses to a single cell occur close in time, for the force calculation of the cell itself as well as for its neighbours. Using the domain decomposition cell scheme, two cell layers have to be kept in the processor cache. For 10000 particles and a typical cell grid size of 20, these two cell layers consume roughly 200 KBytes, which nearly fits well into the L2 cache. Therefore every cell has to be read from the main memory only once per force calculation.

Since ESPResSo does not feature a standard implementation of the link cell scheme, the improvements due to our data organisation in comparison to the conventional

structure can only be be estimated. PolyMD as well as LAMMPS run about 2–3 times slower than ESPResSo for a simple Lennard–Jones liquid, but of course the completely different implementations make it impossible to attribute the speed gain to any special feature. However, the force loop and the integrator are much more optimised in the other codes than in ESPResSo, which makes it unlikely that ESPResSo gains it speed through these program parts.

The other programs choose a higher rate of particle resorts than ESPResSo, which can be easily understood since ESPResSo trades speed at the force evaluation for speed at the resorting. For the Lennard–Jones liquid a Verlet update occurs every 40 time steps in ESPResSo with the optimal setting the skin depth, and every 20 time steps in the other programs.

Another effect of this data organisation is a slightly better readable code. The standard link cell algorithm requires a lot of indirect accesses to the particles through the cell pointers in the particle resorting procedures, which are not needed in ESPResSo. On the other hand the transfer of a particle from one cell to another is not more complicated than the update of a link address in the code, as this is handled by a separate subroutine.

The other two cell systems, namely the N^2 cell system and the layered cell system, are not as efficient as the domain decomposition and only have to be used with certain potentials. The N^2 cell system will calculate the interactions for all particle pairs. This is necessary e. g. for MMM1D, as will be described in chapter 6, or for the calculation of the electrostatic interaction with no periodic boundary conditions, e. g. in the cell model. Since all interactions have to be calculated anyway, a domain decomposition is unnecessary. Instead of this, the particles are load balanced at the beginning of the simulation, i. e. the particle number does not differ by more than one from the average particle number on each node. Once this load balancing is achieved, the particles are not resorted again. This method require only one particle list per node on each node.

The layered cell system is very special and is a combination of the domain decomposition and the N^2–method. The system is split up into cells or layers only along the z–coordinate. Interactions are treated with all particles in the adjacent layers. This cell system probably only makes sense in combination with the MMM2D method as presented in chapter 3.

The concept of cell systems allow for quite different data organisations within ESPResSo, as is needed for some state of the art algorithms. The way that ESPResSo stores the particle data is to our knowledge unique to ESPResSo and highly efficient. In addition the program code is somewhat easier to read instead of more complex. It is easy to add new cell systems if a new algorithm requires a different particle organisation. Therefore this data model fits optimally to our main goals: readable code, state of the art algorithms and of course extensibility.

1.4 The velocity Verlet integrator and the Langevin thermostat

ESPResSo features a velocity Verlet integrator, which is widely used for computer simulations since it is easy to implement, but fast and most importantly does not show

long–time energy drift. Besides the intrinsic NVE ensemble, the integrator can be used for NVT simulations with a Langevin thermostat or a dissipative particle dynamics (DPD) thermostat. For NPT simulations an isotropic box rescaling algorithm exists with an appropriate isotropic thermostat. The integrator can handle particles that are not rotationally invariant, which occur for example in liquid crystal models. Finally systems under shear stress can be simulated using an NEMD integration scheme.

The velocity Verlet integrator

The simulations presented in this work all use the plain velocity Verlet integrator in combination with a Langevin thermostat. Let $r(t)$ denote the positions and $v(t)$ the velocities and $a(t)$ the forces of the particles at time t. If all involved potentials are sufficiently smooth, the velocities can be expanded into an Taylor series around t rsp. $t + \delta t$ and we obtain $v(t + \frac{1}{2}\delta t)$ as:

$$\begin{aligned} v(t + \frac{1}{2}\delta t) &= v(t) + \frac{1}{2}a(t)\delta t + \dot{a}(t + \frac{1}{2}\delta t)\delta t^2 + \mathcal{O}(\delta t^3) \text{ rsp.} \\ v(t + \frac{1}{2}\delta t) &= v(t + \delta t) - \frac{1}{2}a(t + \delta t)\delta t + \dot{a}(t + \frac{1}{2}\delta t)\delta t^2 + \mathcal{O}(\delta t^3)\,. \end{aligned} \tag{1.1}$$

Subtraction of the equations gives

$$\begin{aligned} v(t + \delta t) &= v(t) + \frac{1}{2}\left(a(t) + a(t + \delta t)\right)\delta t + \mathcal{O}(\delta t^3) \\ &= v_{t+\delta t} + \mathcal{O}(\delta t^3)\,, \end{aligned} \tag{1.2}$$

where

$$\begin{aligned} v_{t+\delta t} &:= v_{t+\frac{1}{2}\delta t} + \frac{1}{2}a(t + \delta t)\delta t \text{ and} \\ v_{t+\frac{1}{2}\delta t} &:= v(t) + \frac{1}{2}a(t)\delta t\,. \end{aligned} \tag{1.3}$$

The introduction of the formal half step velocities $v_{t+\frac{1}{2}\delta t}$ is very convenient for the implementation of the algorithm, as we will see in a moment. The positions are obtained similarly as

$$\begin{aligned} r(t + \delta t) &= r(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2 + \mathcal{O}(\delta t^3) \\ &= r_{t+\delta t} + \mathcal{O}(\delta t^3)\,, \end{aligned} \tag{1.4}$$

where

$$r_{t+\delta t} = r(t) + v_{t+\frac{1}{2}\delta t}\delta t\,. \tag{1.5}$$

The presentation above shows that the velocity Verlet algorithm has an numerical error which scales like at least $\mathcal{O}(\delta t^3)$. A slightly more involved derivation shows that the error order in the positions is actually $\mathcal{O}(\delta t^4)$. But since the usual implementation of the Langevin thermostat results in a truncation error of order $\mathcal{O}(\delta t^3)$ as will be discussed below, there is no point in a higher accuracy of the integration scheme. The velocity Verlet algorithm is time–reversible, i. e. inverting the sign of the momenta the algorithm will trace back the positions of the particles up to numerical precision. More

importantly, the algorithm is area preserving, i. e. it will leave the size of any volume element in phase space unchanged [31]. Both time–reversibility and area preservation are necessary for long time energy conservation, which is probably the most important advantage of the velocity Verlet algorithm, especially in NVE simulations.

Using the formal half step velocities, the velocity Verlet algorithm can be implemented such that at no time the position, velocity or force of particle is needed for more than one simulation time step. In pseudo code the propagation from $t = t_0$ to $t = t_1$ with a time step of δt looks like this

$t = t_0$	
calculate accelerations a	
while $t < t_1$	
$v = v + \frac{1}{2}\delta t a$	$v = v_{t+\frac{1}{2}\delta t}$
$r = r + \delta t v$	$r = r_{t+\delta t}$
calculate accelerations a	
$v = v + \frac{1}{2}\delta t a$	$v = v_{t+\delta t}$
$t = t + \delta t$	
end	

The Langevin thermostat

The integrator as presented is only capable of simulating a system in the microcanonical or NVE ensemble. However, the simulations presented later in this work study the behaviour of certain systems in the canonical or NVT ensemble, i. e. at constant temperature. Therefore a thermostat is needed to drive the system into the canonical state, e. g. by introducing artificial degrees of freedom or by coupling the system to a heat bath via stochastic methods [31]. For the simulations a thermostat of the second class was chosen, the Langevin thermostat [38]. Instead of solving Newton's equations of motion, one solves the Langevin equations

$$a_i = -\nabla U((r_i)_i) + \Gamma/m_i v_i + \xi_i(t)/m_i \tag{1.6}$$

where r_i is the position of particle i, v_i its velocity, a_i its acceleration and m_i its mass. Γ is a friction coefficient, and the $x_i(t)$ are Gaussian random variables with

$$\langle \xi_i(t) \rangle = 0 \text{ and } \left\langle \xi_i(t) \cdot \xi_j(t') \right\rangle = 6 k_B T \Gamma \delta_{ij} \delta(t - t') \tag{1.7}$$

i. e. random variables which are uncorrelated both among the particles and in time. The friction term can be interpreted as the dragging of a surrounding viscous medium consisting of considerably smaller molecules, the random term as "kicks" exerted by this medium onto the particles. It can be proven that the Gaussian random variables can be replaced by uniformly distributed ones such as generated by a standard random number generator having the same variance [25].

One tricky point in the combination of the Langevin thermostat with the velocity Verlet integrator arises from the fact that the friction force is velocity dependent.

However, when the forces for time $t + \delta t$ are computed, only the positions belong to time $t + \delta t$, while the velocities still belong to $t + \delta t/2$. It can be shown that this introduces an error of order $\mathcal{O}(\delta t)$ in the forces [17], so the combination of Langevin thermostat and velocity Verlet algorithm has an order of $\mathcal{O}(\delta t^3)$ in the positions and of order $\mathcal{O}(\delta t^2)$ in the velocities.

Another point is even more subtle. The Langevin thermostat evaluates the force at each time step only once, although they are used twice, as $a(t + \delta t)$ and then again in the next time step as $a(t)$. Particularly the random forces ξ_i of the Langevin thermostat are equal for both evaluations. But if the forces are recalculated at the beginning of the integration loop, the random forces will differ, since different random numbers will be drawn. This can happen in ESPResSo because of a crucial change, e. g. adding a single particle or modification of a constraint, since ESPResSo will always recalculate the forces for all particles for the sake of simplicity.

The random contributions enter linearly into $v(t)$ and therefore $r(t)$, so that the two random contributions simply add up. While for using two times the same random value the variance is

$$\left\langle (2\xi_i)^2 \right\rangle = 4 \left\langle \xi_i^2 \right\rangle , \tag{1.8}$$

calculating the random value twice results in the two values being independent and a variance of

$$\left\langle (\xi_i + \xi_i')^2 \right\rangle = \left\langle \xi_i^2 \right\rangle + 2 \left\langle \xi_i \xi_i' \right\rangle + \left\langle \xi_i'^2 \right\rangle = 2 \left\langle \xi_i^2 \right\rangle . \tag{1.9}$$

Therefore the average amplitude of the random forces is reduced by a factor of $\sqrt{2}$, resulting in a lowered temperature. Of course the thermostat will correct for this, however, this may take a few hundred time steps. Simulations, in which critical changes occur at a higher rate, as may happen when bonds are broken and formed anew, will therefore suffer a severe temperature loss.

In ESPResSo this problem is remedied by increasing the amplitude of the random force used when entering the integration loop by a factor of $\sqrt{3}$, so that the total amplitude of the random contribution is the same:

$$\left\langle (\xi_i + \sqrt{3}\xi_i')^2 \right\rangle = \left\langle \xi_i^2 \right\rangle + 2\sqrt{3}\left\langle \xi_i \xi_i' \right\rangle + 3\left\langle \xi_i'^2 \right\rangle = 4\left\langle \xi_i^2 \right\rangle . \tag{1.10}$$

1.5 Short ranged interactions

ESPResSo features several short ranged interactions. In principal two different types of short ranged interactions are possible: nonbonded and bonded interactions. The nonbonded interactions are calculated for all pairs of particles within a certain distance, as described above. The parameters of the interaction depend only on the type of both particles but not on the specific particles. Examples are the Lennard–Jones potential, a Lennard-Jones potential with a cosine tail, the Gay–Berne potential and tabulated potentials given by a freely definable, piecewise linear radial potential.

Bonded interactions occur only between particles for which they were explicitly defined. One example of a bonded interaction is the finite extension nonlinear elastic (FENE) bond, i. e. a simple spring that is normally used as bond in the bead–spring model. Other bonded interactions implemented in ESPResSo are a bond with a harmonic potential and again tabulated bonded potentials. Besides these pair bonds,

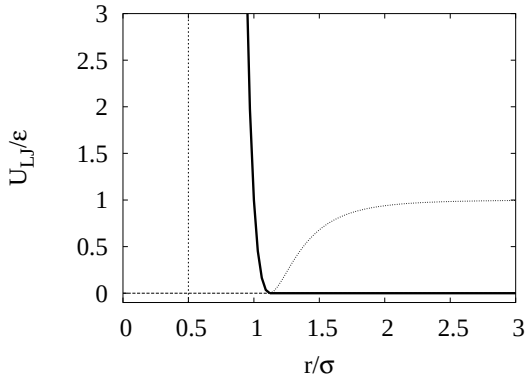


Figure 1.3: The Lennard–Jones potential $U_{LJ}(r)$. The solid black line gives the purely repulsive Lennard–Jones potential, while the dotted curve shows the attractive tail of the plain Lennard–Jones potential. The interaction energy is infinite at $r = 0.5\sigma$, but already at $r = 0.8\sigma$ it is as large as 44ϵ.

in which only two particles occur, ESPResSo also has an angle potential, aiming to keep the angle between three particles at a definable value. ESPResSo can handle bonded interactions of arbitrarily many particles. The implementation of both nonbonded and bonded interactions is well documented and only requires a few trivial additions in the code, so that also more complicated potentials as they arise in atomistic simulations could be implemented easily on demand.

The only short ranged potential that will be used in the simulations presented here is the generalised Lennard–Jones potential, which is given by

$$U_{LJ}(r) = \begin{cases} \infty & \forall r \leq r_0 \\ 4\epsilon \left(\left(\frac{\sigma}{r - r_0} \right)^{12} - \left(\frac{\sigma}{r - r_0} \right)^{6} - \gamma_{shift} \right) & \forall r_0 < r < r_c \\ 0 & \forall r_c \leq r \,, \end{cases} \tag{1.11}$$

where γ_{shift} is chosen such that the potential is continuous in r_c, i. e. $\gamma_{shift} = \left(\frac{\sigma}{r_c - r_0} \right)^{12} - \left(\frac{\sigma}{r_c - r_0} \right)^{6}$. r_0 is called the hard core radius; two particles cannot get closer than this, since the energy at this radius is infinite. $r_0 + \sigma$ is the soft Lennard–Jones diameter, since here the interaction energy is ϵ, which in general is chosen to be $1kT$, i. e. equal to the average thermal energy of a particle. Therefore particles interacting through this potential will not come much closer than $r_0 + \sigma$ in general. r_c is a cutoff to avoid the calculation of the potential at unnecessary large distance, since it decays extremely fast. The potential has one and only one extremum, a minimum at $r = \sqrt[6]{2}$, i. e. the potential has an attractive tail, but is highly repulsive for smaller r. If r_c is finite, the force is only continuous if the potential is cut off in the minimum, in which case the shift is $\gamma_{shift} = -1/4\epsilon$ and the potential is purely repulsive. This is a common choice to model a hard core interaction in a molecular dynamics simulation, and is also called the Weeks–Chandler-Anderson (WCA) potential [99]. Fig. 1.3 shows a typical repulsive Lennard-Jones interaction as is used for the simulations presented here.

In MD simulations, energies are often measured in multiples of the Lennard–Jones energy at contact, ϵ. Lengths are measured in multiples of the Lennard–Jones radius σ. Time finally is measured in multiples of the relaxation time of the Lennard–Jones potential, $\tau_{LJ} = \sigma\sqrt{m/\epsilon}$. This unit system is known as *Lennard–Jones units* and is also used for the simulations presented in chapter 7.

1.6 Long ranged interactions — the electrostatic interaction

The research in the work group developing ESPResSo is focused on polyelectrolytes, i. e. charged polymers. Therefore the calculation of the electrostatic interaction plays an important role in ESPResSo. Despite its simple form the calculation of the plain Coulomb potential is computationally very demanding and can consume up to 60% of the total computation time in a typical simulation. Unlike the short ranged potentials described before, the plain Coulomb potential is long ranged, i. e. in general the interaction cannot be cutoff within the simulation box without severe artifacts, especially in dilute systems. Therefore the Coulomb interaction has to be taken into account for all pairs of charged particles, resulting in an unfavourable $\mathcal{O}(N^2)$ force loop. Although the potential itself is very simple and easy to compute, no more than a few thousand charges can be treated efficiently this way. For the case of a system without periodic boundary conditions this is the only way to calculate the electrostatic interaction in ESPResSo.

With periodic boundary conditions the direct calculation of the electrostatic interaction is impossible already with only a few particles, since many thousands of periodic images need to be considered and deliver considerable contributions. Fortunately, there exist complex, but efficient algorithms for the calculation of the electrostatic interaction in arbitrary periodic boundary conditions. The most common case is periodicity in all three spatial dimensions, which is used in simulations of bulk systems like liquids. Consequently for this kind of periodicity a broad range of highly efficient and well studied algorithms such as the particle–mesh Ewald methods exists. In ESPResSo currently only P3M, a $\mathcal{O}(N \log N)$ mesh–based Ewald method is implemented. This method is well understood in terms both of computational accuracy and efficiency, and fits optimally the systems that are researched using ESPResSo. Chapter 2 presents an overview over a broad range of methods for the calculation of the electrostatic interaction in fully periodic systems.

Periodicity in only two of the three spatial dimensions is appropriate for the simulation of interfaces such as membranes or thin films. These kinds of simulations become more and more important with the growing interest in biological membranes. Until recently, for this kind of periodicity only a few algorithms existed, with a computational efficiency not remotely comparable to the methods available for three dimensional periodicity. This work presents two recently developed algorithms for two dimensionally periodic boundary conditions which are comparable to the methods for three dimensional periodicity, namely MMM2D and ELC. MMM2D uses a convergence factor approach to achieve a computational scaling of $\mathcal{O}(N^{5/3})$ and is suitable for small numbers of particles below 1000 particles, while ELC is not a method itself, but rather utilises parts of MMM2D to allow the use of any method for three dimensional periodicity. The computational order of the ELC correction term is $\mathcal{O}(N)$, so that it does not destroy the scaling of the currently known methods for three dimensional periodicity. Both algorithms are implemented in ESPResSo. They will be discussed to some extent together with an overview over the other methods known in chapters 3 and 4.

For the case of periodicity in only one dimension only few even less efficient algorithms exist. ESPResSo uses MMM1D for this kind of periodicity, a method which is of computational order $\mathcal{O}(N^2)$, but fortunately with a small prefactor. Therefore

MMM1D can still be used for up to 400 particles, which is sufficient for many applications. MMM1D and the possible alternatives will be discussed in chapter 6, an application of MMM1D in chapter 7.

For nonperiodic boundary conditions, ESPResSo uses a trivial pairwise potential summation, which is sufficient for up to 1000 particles due to its simple $1/r$–form. All the algorithms presented so far are methods to calculate the plain Coulomb sum. But for example charges in a solute containing salt interact effectively through an exponentially screened Coulomb potential $e^{-\kappa r}/r$, the Debye–Hückel potential. κ is the inverse screening length and is determined by the salt concentration. ESPResSo also contains such a Debye–Hückel potential to allow simulations of such systems without adding real salt particles. This speeds up simulations at high salt concentrations in two ways: first of all the number of particles is considerably smaller since the salt particles are omitted, and second the screened potential is short ranged, allowing for a calculation by the standard link cell algorithm in a computation time of $\mathcal{O}(N)$.

This section only briefly listed the methods ESPResSo uses for the calculation of the electrostatic interaction. The following chapters will be devoted solely to the presentation of these and other algorithms and discuss their advantages and disadvantages, and the last chapter will bring forward investigations on one of the puzzling effects of the electrostatic interaction, that were performed by the MMM1D implementation in ESPResSo.

1.7 Analysis

A computer simulation itself produces only configurations, i. e. particle positions, velocities and forces. For most applications other analysis tools are needed. Some of these are not only of interest after the simulation is done, but are also useful for checking the consistency of a simulation during runtime. Therefore the analysis tools are integrated into ESPResSo itself and can be called at any time outside or during the integration. One example of such an online analysis is the printing of the measured temperature in the example script in Sec. 1.1. In production simulations writing out the energies can help to detect problems because of a too large time step, for example. One can easily stop the simulation and restart it with more reasonable values without having to wait for the simulation to finish. Except for very time consuming analyses such as the calculation of structure factors, it has proven to be advantageous to calculate most observables online, i. e. during the integration. In ESPResSo this can be done easily, and preliminary results are available long before the simulation actually terminates.

The list of analysis tools is quite long and rapidly growing. Currently ESPResSo can calculate the energy and pressure in detail, or the radial distribution function for arbitrary particle types. For linear molecules ESPResSo can measure the average end–to–end distance r_E, the radius of gyration r_G, the hydrodynamic radius r_H, the average bond length, the internal distance distribution and the mean square displacements g_1 of the particles, g_2 of the particles within their chains and g_3 of the canter of mass. For comparison with scattering experiments ESPResSo is able to calculate *structure factors*. One of the most recent analysis tools in ESPResSo is a fluctuation mode analysis for membrane systems [75].

Analysis tools in ESPResSo are implemented in two different ways: The energy and isotropic pressure calculations are parallelised and handled similarly to the force calculation. All other implementations are not parallelised. Therefore all particle data is first transferred to the master node and then analysed there, allowing for simpler analysis procedures. Since an analysis typically takes place every 1000 time steps or less frequent, the time penalty for this strategy is almost negligible. But the formulation of many of these analyses is much easier on a single data structure, and often the analysis cannot be parallelised efficiently at all.

For the present work only one of the analysis tools was used, the energy calculation, which allows for kind of a quality control. All other presented observables such as radial or azimuthal particle distributions were written directly in Tcl, since they have to take into account the unusual cylindrical geometry of the simulation space. For small numbers of particles, Tcl is efficient enough to perform the analysis in reasonable time, and the analysis can be implemented much easier than plugging it directly into the simulation core.

1.8 Other features

In a simulation nonperiodic dimensions need a constraint, otherwise particles might diffuse away. Constraints in ESPResSo can have arbitrary shape, for example walls, cylinders, or spheres. They have a particle type and interact with the particles through a Lennard–Jones interaction which parameters depend on the particle's type and the constraint's type. This allows for different particles to react differently to the constraint. Writing these constraints is easy, allowing for more exotic constraints like a maze (a system of hollow spheres connected by cylinders).

A plane parallel to the periodic coordinates in a system where the electrostatic interaction is treated via MMM2D can be charged in ESPResSo. Similarly, a rod parallel to the periodic coordinate in a one dimensionally periodic system can be charged, too. Besides the constraints, ESPResSo also allows to directly apply arbitrary external forces to particles, or fix them in some of the spatial coordinates, for example to graft a polymer on a surface. For the simulations presented here only the (charged) cylinder constraint is used.

The integration process can be visualised in real time using the IMD interface, which is used by VMD [45]. VMD is a visualisation tool developed for proteins, but also other molecules including bead–spring models can be displayed. ESPResSo in this case acts as an IMD server. If VMD connects to the simulation, the current positions of the particles are sent to VMD any time the Tcl commando `imd positions` is executed. This handy feature allows to start a simulation and occasionally check visually if everything is going well.

The features of ESPResSo used in the integrator, the cell systems, potentials and constraints, as well as IMD or the analysis are all implemented in C. But some parts of ESPResSo itself are written in Tcl, e. g. most parts of the file I/O, i. e. of the `blockfile` command. This enables the use of the powerful string parsing commands of Tcl and results in highly flexible but still readable functions for the file I/O. Moreover one can easily add parsers for additional block types, such as the "energy" block in the example script, even in the simulation script. Other Tcl helper functions allow

the automated writing of checkpoint blockfiles or the creation of pdb files, a format commonly used by programs for the visualisation of molecules such as VMD or rasmol. A small set of mathematical functions allows basic vector calculations and statistical operations such as calculation of averages and standard deviations or linear regression. Other routines automatically create polymer systems such as networks.

The extensibility of ESPResSo allows for a large community of programmers to add new features. To ensure the compatability of recent changes with the rest of the code, ESPResSo has a *test suite* which contains example scripts for each of the features of ESPResSo. These example scripts assure that ESPResSo is still able to reproduce a result that is assumed to be correct up to numerical errors. Each the scripts is executed on 1-8 nodes, so that the test suite allows to test the full functionality of ESPResSo on a single as well as on many processors.

For an early status report on ESPResSo, see Ref. [8], while a publication describing ESPResSo and its features is currently in progress [9].

2 Calculating the Coulomb interaction under periodic boundary conditions

This chapter presents methods to calculate the electrostatic interaction under periodic boundary conditions. Three aspects of such an algorithm are of special importance, namely the existence of error estimates, the computational scaling, and parallelisation strategies. From time to time methods are put forward for which no error estimates exist and probably cannot be derived at all. Such methods can introduce any kind of errors in a simulation, making the results of the simulation unreliable at best. Moreover, additional accuracy of an algorithm usually causes additional computational effort; error estimates allow to keep the computational effort as small as possible, while still producing reliable results. Therefore methods, for which no error estimates are known nor will never be found to all likeliness, are ignored in the following. Knowing the scaling of the computation time as well as the possibility of parallelisation helps with the choice of the algorithm to be used in a simulation. Most of the techniques described here are highly involved and their implementation might take several weeks of manpower, and one is probably not willing to implement a whole set of different methods and then choose only one of them. An overview of the available methods for the calculation of the electrostatic interaction is therefore useful in finding the optimal algorithm for a specific simulation problem.

The methods for the calculation of the electrostatics can be divided into different classes of algorithms, which each have their advantages and disadvantages. The classes are

The Ewald method The Ewald method dates back to the early 1920's and was first used for the calculation of the electrostatic energy of crystals. The idea of this method is to split the Coulomb potential into two parts, one short ranged, but singular, and a long ranged, but smooth part. The latter part can be treated efficiently in Fourier space. By far the most computer simulations nowadays still use the classical Ewald method for the calculation of the electrostatic interaction. The method can be implemented such that the computational complexity scales as $O(N^{3/2})$. The Ewald method is the method of choice for systems with low accuracy requirements and less than 500 particles. Because of the Fourier space calculations the standard Ewald method cannot be trivially extended to other boundary conditions; one has to go through the full derivation of the method again, and the scaling may be worse, see chapters 3 and 6. One disadvantage of all Ewald type methods is that an artificial splitting parameter α is introduced, on which the computational accuracy depends crucially, and therefore has to be chosen very carefully. Moreover increasing the accuracy is computationally expensive.

Mesh–based Ewald methods A better scaling than the plain Ewald method can be

obtained if the Fourier space part is calculated using fast Fourier transforms. To this aim the charges have to be interpolated onto a mesh. There are different ways to do this, and consequently different methods can be obtained, namely the particle–particle particle–mesh method (P3M), the particle–mesh Ewald (PME) and the smooth PME (SPME). They all scale like $O(N \log N)$ and are faster than the plain Ewald method for more than about 1000 particles, but the other drawbacks of the Ewald methods, namely the crucial parameter α and the bad scaling with the error estimate, also apply to the mesh–based Ewald methods. Nevertheless for low accuracy requirements and 1000 to 10^6 particles, a mesh–based Ewald method is optimally suited.

Multipole methods These methods are based on a multipole expansion of the electrostatic interaction, allowing for a product decomposition. Since the multipole expansion is calculated in real space, the periodicity only enters through the coefficients in the expansion. Therefore multipole methods are not restricted to a special kind of periodicity as are other kinds of methods. Depending on the implementation the computational order of these methods is $\mathcal{O}(N \log N)$ (Barnes–Hut tree codes) or $\mathcal{O}(N)$ (fast multipole method, FMM). Increasing the error bounds is not expensive, but these methods suffer from large prefactors and are only competitive for particle numbers above 10^6.

Lekner method These methods are based on a screening approach, i. e. the electrostatic interaction is calculated analytically for an exponentially screened Coulomb interaction, then the limit of infinite screening length is considered. This approach leads to results which differ from the results of the Ewald methods or the FMM by some multiple of the square of the dipole moment. Again the situation is different for partially periodic systems, as we will see in the following chapters. The Lekner method has a computational scaling of $\mathcal{O}(N^2)$ and is therefore suitable only for small numbers of particles. One advantage is that the computational error is proportional to the logarithm of the error bound. For small numbers of particles and high accuracy goals the Lekner method may be optimal.

MMM Sperb and Strebel modified the Lekner sum such that in three dimensional periodic boundary conditions a scaling of $\mathcal{O}(N \log N)$ can be achieved, with a prefactor similar to the mesh–based Ewald methods. This method was termed MMM. MMM maintains the favourable scaling with the logarithm of the error bound. For high accuracy goals and 1000 to 10^6 particles, MMM is the optimal method.

Two other algorithms, that are still under development and therefore are not mentioned in the list above, are the local algorithm for Coulomb interactions by T. Maggs [61] and the finite difference scheme method of I. Tsukerman [96]. The Maggs method is based on the introduction of a propagating field E, which is nothing but a discretised version of the electric field. For this field the propagation rules are only local, so that the computation time scales with the number of grid points and should therefore be linear with the number of particles. The method has two advantages: it allows for a locally changing dielectric constant, which is impossible to achieve with

the other approaches presented here, and it integrates nicely into other lattice methods such as lattice–Boltzmann. Because of the complex physical background of the method so far no error estimates exist. Currently, the Maggs method is implemented into ESPResSo.

The approach of Igor Tsukerman is based on solving the Poisson equation for the simulation box on a grid. To this aim, the singular parts close to the point charges are taken out and solved analytically, while the rest of the potential is solved by using a finite difference scheme on the grid. Finite difference schemes are local methods, making this approach computationally efficient. Errors arise only from the interpolation of the potential on a grid and from the finite difference scheme, for which error estimates exist. Like the Maggs method, this method takes advantage of the fact that in a MD simulation a good approximation of the current electrostatic potential is given simply by the potential of the last time step. Finite difference schemes solve the differential equation iteratively, so having a good starting point reduces the computation time considerably. The method of Tsukerman has not yet been implemented, but it is planned to include it into ESPResSo.

This chapter and the two following try to give an overview over the most widely used methods for both fully and partially periodic boundary conditions. Besides the algorithmic and mathematical details of these methods, also some mathematical aspects of the calculation of the electrostatic interaction are discussed. The Coulomb sum is mathematically more complex than its simple formulation suggests. The main reason for this is that the Coulomb sum is only conditionally convergent so that the order of summation is important. This restricts the mathematical arsenal drastically since many mathematical devices require absolute convergence, and is the source of many other unusual properties of the electrostatic interaction in periodic boundary conditions, for example the electrostatic interaction depends on the dielectric constant surrounding the infinite (!) periodic system.

This chapter gives an overview over the methods that can be used in three dimensionally periodic boundary conditions. The first section of this chapter deals with the general prerequisites and the strict mathematical definition of the problem. The second section will shortly recall the details of some of the Ewald type methods, especially of P3M, which is used in ESPResSo. The next sections treat multipole methods and the Lekner type methods. The final section is devoted to MMM, which is the foundation of the methods ESPResSo uses for partially periodic systems and are presented in the following two chapters.

2.1 General prerequisites

We consider a system of N particles with charges $q_i \in \mathbb{R}$ and pairwise different coordinates $p_i = (x_i, y_i, z_i)^T \in B_0$, $i = 1, \ldots, N$, where

$$B_0 = \left(-\frac{\lambda_x}{2}, \frac{\lambda_x}{2}\right] \times \left(-\frac{\lambda_y}{2}, \frac{\lambda_y}{2}\right] \times \left(-\frac{\lambda_z}{2}, \frac{\lambda_z}{2}\right] \tag{2.1}$$

is the primary simulation box. Furthermore it is assumed that the system is charge

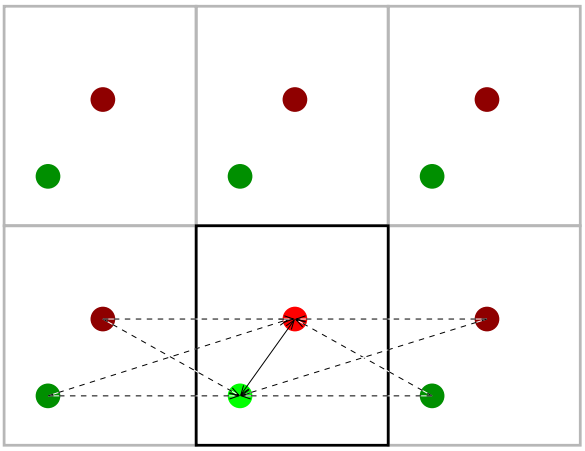


Figure 2.1: Schematic representation of the periodically replicated system. The black box is the primary box, the grey boxes are the images. The arrows denote the interactions that have to be taken into account in the lower three boxes. The interaction shown with a solid line is taken into account with weight 1, the dashed lines have weight 1/2. For the upper three boxes no other interactions occur.

neutral, i. e.

$$\sum_{i=1}^{N} q_i = 0\,. \tag{2.2}$$

Now we replicate this system regularly along all spatial dimensions, as shown in Fig. (2.1) for a two dimensional system. Since we treat the primary box as our "probe", one has to take care that only the energies associated with the particles inside the primary box are considered. The energy of the primary simulation box divided by its volume then gives the average energy density of the full infinite system, which is what one is interested in.

The interactions of two particles in the primary simulation box are taken into account with a weight of 1. Interactions between a particle of the primary simulation box and a particle in a periodically replicated box are taken into account with a weight of 1/2, as only the particle of the primary simulation box belongs to the probe region. The interactions between particles in two periodically replicated boxes are ignored.

We introduce some notations that will be used frequently in the following. The real space lattice vectors are

$$n_{klm} := (k\lambda_x, l\lambda_y, m\lambda_z)^T, n_{kl} := n_{kl0} \quad \text{and} \quad n_k := n_{k0} \tag{2.3}$$

for $k, l, m \in \mathbb{Z}$. The inverse box dimensions are

$$u_x = \frac{1}{\lambda_x}\,, \quad u_y = \frac{1}{\lambda_y}\,, \quad \text{and} \quad u_z = \frac{1}{\lambda_z}\,. \tag{2.4}$$

The relative particle positions are

$$p_{ij} = (x_{ij}, y_{ij}, z_{ij}) = p_i - p_j\,. \tag{2.5}$$

For $r = (x, y, z)$ the distances from the origin to the (k, l, m)–th periodic image are

$$\begin{aligned} r_{klm} = |r + n_{klm}| = \sqrt{(x + k\lambda_x)^2 + (y + l\lambda_y)^2 + (z + m\lambda_z)^2}\,, \\ r_{kl} = r_{kl0}\,, \quad r_k = r_{k0} \quad \text{and} \quad r = r_0\,, \end{aligned} \tag{2.6}$$

and the yz–plane distances are

$$\rho_{lm} = \sqrt{(y + l\lambda_y)^2 + (z + m\lambda_z)^2}\,, \quad \rho_l = \rho_{l0}\,, \quad \text{and} \quad \rho = \rho_0\,. \tag{2.7}$$

Using these notations, the electrostatic problem in periodic boundary conditions is given by

$$E = \frac{1}{2} \sum_{(k,l,m\in\mathbb{Z}^3} \sum_{i,j}{}' \frac{q_i q_j}{|p_{ij} + n_{klm}|}\,.$$

The prime at the inner sum denotes that the summand for $i = j$ has to be omitted for $(k, l, m) = 0$. Although this formula looks fine, it is not a strict mathematically definition. The reason is that the sum is only conditionally convergent, so the order of summation has to be defined. Usually a spherical limit is applied, i. e.

$$E = \frac{1}{2} \sum_{S=0}^{\infty} \sum_{k^2+l^2+m^2=S} \sum_{i,j}{}' \frac{q_i q_j}{|p_{ij} + n_{klm}|}\,. \tag{2.8}$$

In chapter 4 we will have to deal with a different order of summation. If one adds up the particles along z slabwise, i. e. ordered by increasing z–distance, but radially in x and y, Smith has shown [88] that

$$-\frac{1}{2} \sum_{\mu=0}^{\infty} \sum_{m=\pm\mu} \sum_{S=0}^{\infty} \sum_{k^2+l^2=S} \sum_{i,j=1}^{N}{}' \frac{q_i q_j}{|p_i - p_j + n_{klm}|} = E + 2\pi M_z^2 - \frac{2\pi M^2}{3}\,. \tag{2.9}$$

where

$$M = (M_x, M_y, M_z) = \sum_{i=1}^{N} q_i p_i \tag{2.10}$$

is the net dipole moment of the primary simulation box. We will see below that the Ewald–sum contains a summand $2\pi M^2/3$, which is just exchanged by $2\pi M_z^2$ by this change of order.

The term $2\pi M^2/3$ is called the *dipole term*. Besides determining the order of summation, it has another interesting property. The spherical summation order is equivalent to the limit of a large, spherically bounded regular grid of images of the simulation box embedded in vacuum, basically a crystalline ball (although for many particles the "crystal" may be quite complex). If the surrounding space is filled by a homogenous medium with a dielectric constant ϵ', the particles of the ball will feel a polarisation force. It can be shown that this leads to an additional contribution that will not vanish even in the limit of an infinite ball, even though then the complete space is filled by copies of the simulation box. The additional contribution for the infinite ball is again given by a modification of only the dipole term, which then reads

$$\frac{2\pi M^2}{2\epsilon' + 1}\,. \tag{2.11}$$

This again shows that even in the limit of a fully filled space the Coulomb sum "remembers" the way the summation was done, which is of course only possible due to its conditional convergence.

2.2 The Ewald method

Ewald developed this method in the early 1920's to analytically calculate the energy contained in simple salt crystals [29]. The basic idea in deriving the Ewald sum is to split the potential into a short ranged part and a long ranged part by using

$$\frac{1}{r} = \frac{\mathrm{erf}(\alpha r)}{r} + \frac{\mathrm{erfc}(\alpha r)}{r}, \tag{2.12}$$

where

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

is the error function and

$$\mathrm{erfc}(x) = 1 - \mathrm{erf}(x)$$

its complement. α is the Ewald splitting parameter and can be chosen arbitrarily. The complementary error function drops exponentially with increasing x, so that by choosing α properly the contribution of the second summand is negligible even for distances much smaller than the box length. The first, long ranged part on the other hand still has to be evaluated over long distances, but is now smooth everywhere, especially at 0, so that it is accessible to a broader range of transformations. The Ewald formula is obtained by performing a three dimensional Fourier transform on the long ranged part. Because the long ranged part of the potential sum is smooth everywhere, the Fourier sum is very well convergent and allows for a small cutoff. The final result is given by

$$\begin{aligned} E =& \frac{1}{2} \sum_{i,j} {\sum_{k,l,m \in \mathbb{Z}^3}}' q_i q_j \frac{\mathrm{erfc}(\alpha |r + n_{klm}|)}{|r + n_{klm}|} + \\ & \frac{1}{2\pi \lambda_x \lambda_y \lambda_z} \sum_{(p,q,r) \in \mathbb{Z}^3} \sum_{i,j} q_i q_j \frac{4\pi^2}{k_{pqr}^2} e^{-\frac{k_{pqr}^2}{4\alpha^2}} \cos(k_{pqr} p_{ij}) - \\ & \frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^{N} q_i^2 + \frac{2\pi M^2}{3} \end{aligned} \tag{2.13}$$

where $k_{pqr} = (u_x p, u_y q, u_z r)$.

The first sum is called the *real space sum* and accounts for the short ranged part of the split Coulomb potential. The second sum is called the *Fourier space sum* as it accounts for the long ranged part in the Fourier space, and the third summand is called the *self energy*, which accounts for the contribution of the long range part for $i = j$ in the primary simulation box. This contribution is artificially added to allow the application of the Fourier transform and therefore has to be subtracted again. The last term of Eqn. (2.13) is the dipole term as discussed in the previous section. Note that the right hand side does not dependent on α, since the left hand side is independent of α.

The Ewald summation formula has two computational advantages. First of all, the involved sums are all well convergent and can therefore be cut off to obtain an affordable computational effort. Second, the far formula can be evaluated in linear

time with respect to N by means of a product decomposition, as can be seen from the addition theorem for the cosine, which leads to

$$\begin{aligned}
&\frac{1}{2\pi\lambda_x\lambda_y\lambda_z}\sum_{(p,q,r)\in\mathbb{Z}^3}\sum_{i,j}q_iq_j\frac{4\pi^2}{k_{pqr}^2}e^{-\frac{k_{pqr}^2}{4\alpha}}cos(k_{pqr}p_{ij}) = \\
&\frac{1}{2\pi\lambda_x\lambda_y\lambda_z}\sum_{(p,q,r)\in\mathbb{Z}^3}\frac{4\pi^2}{k_{pqr}^2}e^{-\frac{k_{pqr}^2}{4\alpha}}\left(\left(\sum_i q_i cos(k_{pqr}p_i)\right)\left(\sum_j q_j cos(k_{pqr}p_j)\right)+\right. \\
&\qquad\left.\left(\sum_i q_i sin(k_{pqr}p_i)\right)\left(\sum_j q_j sin(k_{pqr}p_j)\right)\right).
\end{aligned} \tag{2.14}$$

One only has to calculate the sums over the cosines rsp. sines for each (p,q,r)–vector, which can be clearly done in linear time. The advantage of such a product decomposition is not only the foundation for the improved computational scaling of the Ewald sum, but also for many other methods like the multipole methods and the MMM algorithm.

Still, the evaluation of the near formula needs an N^2–loop. Now the unspecified parameter α comes into play. It can be shown [48], that the RMS (root mean square) force error of the near formula will be constant if the cutoff radius is chosen proportional to $1/\alpha$, while the Fourier space cutoff has to be chosen proportional to α. This leads to an computational effort of $\mathcal{O}(N^2/\alpha^3)+\mathcal{O}(N\alpha^3)$, which is minimal for $\alpha\sim N^{1/6}$, in which case the overall scaling is $\mathcal{O}(N^{3/2})$. α has to grow with increasing N, as this basically puts more and more effort from the real space into the computationally favourable Fourier space. Therefore one can safely retreat to the assumption that the cutoff of the near formula is smaller than the box lengths, such that each particle interacts with no more than one copy of each other particle, i. e. a minimum image convention can be applied. In practical applications the cutoff will be even smaller, such that for an efficient implementation tricks like the link cell algorithm can be used.

Although the parameter α is crucial for the tuning of the Ewald summation, it also has a severe drawback: If α is not chosen properly with respect to the other parameters, the computational *accuracy* may drop considerably. Moreover the computational effort to calculate the Ewald sum grows rapidly with increasing accuracy requirements. Therefore accurate error estimates are required, which were given by Kolafa and Perram in 1992 [48], and can be used to tune α and the real and Fourier space cutoffs.

2.3 A mesh–based Ewald method — P3M

To speed up the Ewald summation even further, one can improve the computation speed of the far formula by using a fast Fourier transform. This requires the charges to be located on a regular grid, which is obviously not the case in general. Therefore the charges are replaced by a charge distribution on a regular grid. For this interpolation of the discrete charges onto this grid several possibilities exist like splines or Lagrange interpolation. The forces are obtained from the energy by differentiation, which can be done either in real space or in the Fourier space. These choices lead to several FFT

based extensions of the Ewald summation, although not all combinations are possible due to mathematical restrictions.

The particle mesh Ewald method (PME) uses a Lagrange interpolation scheme combined with an Fourier space differentiation [15]. The smooth particle mesh Ewald method (PME) uses cardinal B–splines and an analytical differentiation in real space [28]. The P3M method of Hockney and Eastwood [43] is the oldest one and uses cardinal B–splines and a Fourier space numerical real space differentiation on the neighbouring grid points. In [19] one can find a detailed discussion of the different methods and the involved subtleties. It turns out that the oldest method, P3M, is actually the best one to use. ESPResSo uses the P3M method, but instead of the numerical real space differentiation, a Fourier space differentiation is used.

To understand the basic principles of all the mesh–based Ewald methods, we define $\tilde{g}(k) = 4\pi/k^2$, which is nothing but the Fourier transformed Green's function of the Coulomb potential, $\tilde{\gamma}(k) = e^{-k^2/(4\alpha^2)}$, and $\tilde{\rho}(k)$ as the Fourier transformed charge distribution ρ. Then the Fourier space energy can be rewritten as

$$E_{Fourier} = \frac{1}{2\lambda_x\lambda_y\lambda_z} \sum_j q_j \sum_{k\neq 0} \tilde{g}(k)\tilde{\gamma}(k)\tilde{\rho}(k)e^{ikr_j} .$$

I. e. the energy can be obtained by Fourier transforming the charge distribution, multiplication with the function $G = \tilde{g}(k)\tilde{\gamma}(k)$ and transforming back. In other words, the energy is actually the Coulomb energy of ρ convoluted with $\gamma(k)$. Since γ is a Gaussian, ρ is basically smeared out. As said before, the evaluation of the formula can be accelerated considerably by the use of fast Fourier transforms for charges located on a regular grid. All grid methods more or less adhere to this scheme, although G is chosen differently. The reason for this is that by shifting onto a grid the continuum Green's function is no longer optimal with respect to the obtained force error due to the interpolation [19]. Hockney and Eastwood calculated the optimal influence function G and give its Fourier transform as

$$\tilde{G}(k) = \frac{\tilde{D}(k)\sum_{m\in\mathbb{Z}^3} \tilde{U}^2(k+\frac{2\pi}{h}m)\tilde{R}(k+\frac{2\pi}{h}m)}{|\tilde{D}(k)|^2\left(\sum_{m\in\mathbb{Z}^3}\tilde{U}^2(k+\frac{2\pi}{h}m)\right)^2} ,$$

where $\tilde{D}(k)$ denotes the differentiation operator used, e. g. $-\mathrm{i}k$ in the case of ESPResSo, $\tilde{U} = \tilde{W}(k)/V_m$ are the Fourier transformed interpolation coefficients divided by the mesh cell size, and $\tilde{R}(k) = -\mathrm{i}k\tilde{g}(k)\tilde{\gamma}(k)$ is the Fourier transform of the exact force.

In pseudo code, the force evaluation looks as follows, assuming that $\tilde{G}(k)$ and $\tilde{D}(k)$ have been calculated once before the integration starts:

interpolate charges onto the grid
Fourier transform ρ
multiply by $\tilde{G}(k)$
for each coordinate d
multiply by $\tilde{D}_d(k)$
Fourier transform back to obtain F_d

Hence for the forces a total of four Fourier transformations are needed. Using the isometry of the Fourier transform, the energy calculation can be performed with only the Fourier transformation of the charge distribution.

Although the mesh–based Ewald methods are mathematically much more involved compared to the classical Ewald sum, and new sources of errors arise from the interpolation of the charges, rigorous error bounds for all methods exist [78, 43, 48]. These error estimates are computationally unwieldy, but the error depends as crucial on α as for the plain Ewald method, therefore using these error estimates to tune the parameters is mandatory. Quite some effort has been put into the efficient evaluation of the error measures [20]. The dependency of the computation time on the required accuracy is even worse compared to the Ewald method, and increasing the accuracy above a relative RMS force error of about 10^{-5} is impossible within reasonable computational effort.

2.4 The fast multipole method

The key to the improved scaling of the Ewald method is a product decomposition of the Fourier space sum. The multipole methods are based on a product decomposition in real space. Let $x = (r, \theta, \phi)$ and $y = (r', \theta', \phi')$ be the spherical coordinates of two points, $r' < r$. Let $\gamma = \angle(x, y)$ be the angle between the origin and x and y. Then $|x - y| = r^2 + r'^2 - 2rr' \cos\gamma$ and $\cos\gamma = \cos\theta\cos\theta' + \sin\theta\sin\theta'\cos(\phi - \phi')$. The *multipole expansion* of $1/|x - y|$ is then given by

$$\frac{1}{|x-y|} = \sum_{n=0}^{\infty} \frac{r'^n}{r^{n+1}} P_n(\cos\gamma)\,, \tag{2.15}$$

where the P_n are the Legendre polynomials. This is not yet a product decomposition since γ is a mix of coordinates of both points. We separate them using the addition theorem for the Legendre polynomials

$$P_n(\cos\gamma) = \sum_{m=-n}^{n} Y_n^{-m}(\theta', \phi') Y_n^m(\theta, \phi)\,,$$

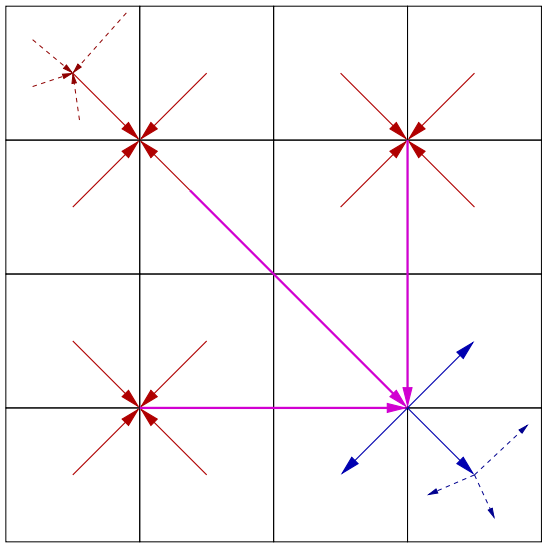
where Y_n^m are the surface harmonics of the first kind, see [1]. The final product decomposition is given by

$$\frac{1}{|x-y|} = \sum_{n,m} \left(Y_n^{-m}(\theta', \phi') r'^n \right) \left(\frac{Y_n^m(\theta, \phi)}{r^{n+1}} \right), \tag{2.16}$$

Because of the requirement $r' < r$, we cannot use a single origin for all interactions, and also the interactions of particles close together have to be calculated directly because of the bad convergence of the multipole expansion. This problem is similar to the problem of the short ranged forces, where only the interactions with close by cells have to be treated. In the case of the multipole method, we treat all cells not adjacent using the multipole expansions, while the adjacent cells are treated using the standard pairwise Coulomb sum. But to gain anything from the multipole expansion, we have to be able to combine the multipole expansion from the remote cells to a

multipole expansions of larger clusters. This requires a procedure to shift the origin of a multipole expansion, which can be achieved by simple linear recombinations of the coefficients $(Y_n^{-m}(\theta', \phi')r'^n)$ and is called a *translation* operation. During the expansion process, it is advantageous to be able to exchange the roles of the particles during a shift, i. e. translate the origin of the expansion from the vicinity of one of the particles to the other, which is similarly possible. This procedure is called a *conversion* operation.

Now let us assume that the number of cells per simulation box side is a power of two, $S = 2^L$, where L is termed the number of *levels*. Always 8 of the cells of the lowest level are combined to form a division of the simulation box into 2^{L-1} larger cells per side, and so on. We can combine the multipole expansions of eight neighbouring of level L cells through translations into one multipole expansions for the cell of level $L-1$ formed by the eight original cells. This procedure can be continued again to obtain multipole expansions for all cells on each level. Once we have arrived at the bottom level $L = 1$ we convert the expansion to local expansions in all three other top level cells and distribute them up to the higher levels again until for all cells the multipole expansion of the full system is available. The multipole expansion for the top level are used to calculate the electrostatic interaction. Graphically this looks like this:



The dotted lines represent the calculation of the multipole expansion and the calculation of the energy rsp. the forces, the solid lines translations and conversions. Here only the data flow from one cell to another is shown, in a real simulation this flow occurs for all pairs of cells, and of course the number of cells is much larger. In pseudo code the energy rsp. force calculation looks like this:

for each particle i
add contribution of particle p **to the local multipole expansion of its cell**
end
for each $l = L \dots 2$
for each cell c **of level** l
add multipole expansion to expansion of the level $l-1$ **supercell, translated to its center**
end
for each cell c **of level** 1
for each cell c' **of level** 1
add converted contribution of cell c **to the local expansion for cell** c'
for each $l = 2 \dots L$
for each cell c **of level** l
calculate local multipole expansion from the translated expansions of same level neighbors and of the level $l-1$ **supercell**
end

The algorithm as presented here is called the fast multipole method [57,56,36]. The number of terms needed in the multipole expansion only depends on the precession requirement and the number of particles in a cell. At constant density and constant number of particles per cell the number of operations therefore only depends on the loops shown above, which are all either of order N or $S^3(1+1/8+1/64+\dots) = \mathcal{O}(N)$, such that the overall computational order is $\mathcal{O}(N)$. One drawback of the method is that all the intermediate multipole expansions have to be stored, since they are needed in the last loop. This can use a considerable amount of memory, since the number of terms in the multipole expansion can be large.

Tree codes or Barnes–Hut methods [11] work similar with respect to the generation of the multipole expansions, but use a different algorithm for spreading them to the other cells. This algorithm is not strictly hierarchically and therefore reduces the computational order to $\mathcal{O}(N \log N)$, but considerably less memory is consumed. The algorithms presented so far are only suitable for the nonperiodic case, but modifications for arbitrary periodic boundary conditions exist in terms of modified coefficients of the multipole expansion.

2.5 The Lekner sum

Both the fast multipole methods and the Ewald type of methods adhere to the spherical summation order for the electrostatic interaction. In contrast to this, Lekner came up with a different formula [55], which is based on transforming the force sum along only one coordinate into a rapidly converging sum. Naturally, the summation order

is rodwise, along the transformed coordinate first. This alone is no problem, since the difference to the spherical limit is given by a multipole of the dipole moment, very similar to the slabwise summation [87]. Also the change to forces instead of energy is problematical, since it is equivalent to changing the order of differentiation and summation in the only conditionally convergent Coulomb sum. Nevertheless the Lekner summation has been used frequently in recent computer simulations [60,65,90], mainly because of its extremely good convergence, which makes it very favourable for small numbers of particles.

We treat the summation along the z axis as a function of z and expand this function in a Fourier series. This results in the force formula

$$\tilde{F}_i = 8\pi \sum_{j\neq i} q_i q_j \sum_{l,m\in\mathbb{Z}} \sum_{p\geq 1} p\, \mathrm{K}_0\left(2\pi u_x p \rho_{lm}\right)\ cos(2\pi u_x p x)\,, \tag{2.17}$$

The tilde on the forces denotes that this force is different from the forces obtained by using the spherical summation limit. The representation is nevertheless well convergent, at least if all charges q_j are well separated in y and z from q_i, i. e. if ρ_{lm} is large enough for all l, m. By spatial symmetry we can choose the coordinate of closest approach as the x–coordinate. For particles close together in all coordinates one has to employ a different method to calculate the electrostatic interaction, see for example Sperb [90], who suggests the following formula

$$\begin{aligned}\tilde{F}_i = 8\pi \sum_{j\neq i} q_i q_j \sum_{l,m\in\mathbb{Z}} \Bigg(& \frac{z}{\left(\rho_{lm}^2 + z^2\right)^{\frac{3}{2}}} - \\ & \sum_{k=0}^{\infty} \binom{-\frac{3}{2}}{k} r^{2k} \left(\psi^{2k+p}(1+z) - \psi^{2k+p}(1-z)\right)\Bigg)\,.\end{aligned} \tag{2.18}$$

This version in contrast only converges for small ρ_{lm}, but again very fast. Like we can choose the Fourier transformed coordinate freely in Eqn. (2.17), we also can choose between inner sum of Eqn. (2.17) and the alternative form of Eqn. (2.18), depending on the distance of the particles. Therefore for all values of ρ_{lm} a well convergent formula exists. For particles sufficiently far away, Eqn. (2.17) is used, while for particles too close together Eqn. (2.18) is used. Consequently the formulas are called *far* rsp. *near* formula in the following. This concept is also used in the MMM, MMM2D, MMM1D and the ELC methods presented in the following.

For the Lekner method upper error bounds can be given easily since the Bessel functions drop essentially exponentially fast, allowing for a simple approximation of the sum by an integral. This error estimates are much less sharp than the error estimates for the Ewald type methods, but here the error bound only enters logarithmically into the computation time, so that excessive accuracy has only small impact on the overall performance.

2.6 MMM

Yet another approach to tackle the conditionally convergent Coulomb sum is used for MMM[1]. Instead of defining the summation order, one can also multiply each summand

[1] Even the authors of the method have no idea what this acronym stands for.

by a continuous factor $c(\beta, r_{ij}, n_{klm})$ such that the sum is absolutely convergent for $\beta > 0$, but $c(0, ., .) = 1$. The energy is then defined as the limit $\beta \to 0$ of the sum, i. e. β is an artificial convergence parameter. For a convergence factor of $e^{-\beta n_{klm}^2}$ the limit is the same as the spherical limit, and one can derive the classical Ewald method quite conveniently through this approach [87]. To derive the formulas for MMM, one has to use a different convergence factor, namely $e^{-\beta|r_{ij}+n_{klm}|}$, which defines the alternative energy

$$\tilde{E} = \frac{1}{2} \lim_{\beta \to 0} \sum_{k,l,m \in \mathbb{Z}} \sum_{i,j=1}^{N}{}' \frac{q_i q_j e^{-\beta|p_{ij}+n_{klm}|}}{|p_{ij}+n_{klm}|} =: \frac{1}{2} \lim_{\beta \to 0} \sum_{i,j=1}^{N} q_i q_j \phi_\beta(x_{ij}, y_{ij}, z_{ij}) \,. \tag{2.19}$$

ϕ_β is given by

$$\phi_\beta(x,y,z) = \tilde{\phi}_\beta(x,y,z) + \begin{cases} \frac{e^{-\beta r}}{r} & (x,y,z) \neq (0,0,0) \\ 0 & (x,y,z) = (0,0,0) \end{cases} \tag{2.20}$$

where

$$\tilde{\phi}_\beta(x,y,z) = \sum_{(k,l,m)\neq 0} \frac{e^{-\beta r_{klm}}}{r_{klm}} \,. \tag{2.21}$$

The limit $\tilde{E}$ exists, but again differs by some multiple of the square of the dipole moment from the spherical limit as obtained by the Ewald summation [87]. From the physical point of view the Coulomb interaction is replaced by a screened Coulomb interaction with screening length $1/\beta$. $\tilde{E}$ is then the energy in the limit of infinite screening length. But because of the conditional convergence of the electrostatic sum, this is not necessarily the same as the energy of an unscreened system. Since the difference to the Ewald methods only depends on the dipole moment of the system, the correction can be calculated easily in linear time and can be ignored with respect to accuracy as well as to computation time.

Starting from this convergence factor approach, R. Strebel and R. Sperb constructed a method of computational order $\mathcal{O}(N \log N)$, MMM [92]. The favourable scaling is obtained, very much like in the Ewald case, by technical tricks in the calculation of the far formula. The far formula has a product decomposition and can be evaluated hierarchically similarly to the fast multipole methods.

For particles sufficiently separated in the z–axis one can Fourier transform the potential along both x and y. We obtain the far formula as

$$\begin{aligned} \phi(x,y,z) = u_x u_y \sum_{p,q\neq 0} \frac{e^{2\pi f_{pq} z} + e^{2\pi f_{pq}(\lambda_z - z)}}{f_{pq}\left(e^{2\pi f_{pq}\lambda_z} - 1\right)} e^{2\pi i u_y q y} e^{2\pi i u_x p x} + \\ 2\pi u_x u_y \left(u_z z^2 - z + \frac{\lambda_z}{6}\right) , \end{aligned} \tag{2.22}$$

where

$$\begin{aligned} f_{pq} = \sqrt{(u_x p)^2 + (u_y q)^2} \,, \quad f_p = u_x p \,, \quad f_q = u_x q \,, \\ \omega_p = 2\pi u_x p \quad \text{and} \quad \omega_q = 2\pi u_y q \,. \end{aligned} \tag{2.23}$$

The advantage of this formula is that it allows for a product decomposition into components of the particles. For example

$$e^{2\pi f_{pq} z} = e^{2\pi f_{pq}(z_i - z_j)} = e^{2\pi f_{pq} z_i} e^{-2\pi f_{pq} z_j}$$

etc. Therefore one just has to calculate the sum over all these exponentials on the left side and on the right side and multiply them together, which can be done in $\mathcal{O}(N)$ computation time. As can be seen easily, the convergence of the series is excellent as long as z is sufficiently large. By symmetry one can choose the coordinate with the largest distance as z to optimise the convergence. Similar to the Lekner sum, we need a different formula if all coordinates are small, i. e. for particles close to each other. For sufficiently small $u_y\rho$ and $u_x x$ we obtain the near formula as

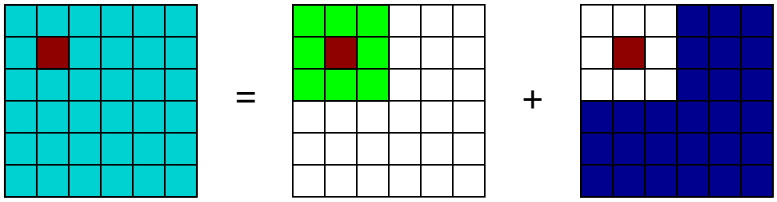
$$\begin{aligned}\tilde{\phi}(x,y,z) = & 2u_x u_y \sum_{p,q>0} \frac{\cosh(2\pi f_{pq} z)}{f_{pq}\left(e^{2\pi f_{pq}\lambda_z} - 1\right)} e^{2\pi i u_y q y} e^{2\pi i u_x p x} + \\ & 4u_x \sum_{l,p>0} \left(\mathrm{K}_0(2\pi u_x p \rho_l) + \mathrm{K}_0(2\pi u_x p \rho_{-l})\right) cos(2\pi u_x p x) - \\ & 2u_x \sum_{n\geq 1} \frac{b_{2n}}{2n(2n)!} \mathrm{Re}\big((2\pi u_y (z+iy))^{2n}\big) + \\ & u_x \sum_{n\geq 0} \binom{-\frac{1}{2}}{n} \frac{\left(\psi^{(2n)}(1+u_x x) + \psi^{(2n)}(1-u_x x)\right)}{(2n)!} \rho^{2n} - \\ & 2\log(4\pi)\,.\end{aligned} \tag{2.24}$$

Note that this time we calculate $\tilde{\phi}$ instead of ϕ, i. e. we omit the contribution of the primary simulation box. This is very convenient as it includes the case of self energy and makes $\tilde{\phi}$ a smooth function. To obtain ϕ one has to add the $1/r$ contribution of the primary box. The self energy is given by

$$\begin{aligned}\tilde{\phi}(0,0,0) = & 2u_x u_y \sum_{p,q>0} \frac{1}{f_{pq}\left(e^{2\pi f_{pq}\lambda_z} - 1\right)} + 8u_x \sum_{l,p>0} \mathrm{K}_0(2\pi u_x \lambda_y p l) + \\ & 2u_x \psi^{(0)}(1) - 2\log(4\pi)\,.\end{aligned} \tag{2.25}$$

Eqn. (2.24) is derived using the same convergence factor approach as used for Eqn. (2.22), and consequently the same singularity in β is obtained. This is important since otherwise the charge neutrality argument does not hold and the limit $\beta \to 0$ could not be performed.

A simple implementation segments the simulation box in $B = S^3$ smaller boxes or cells. For all particles the interactions within the cell itself and the 26 neighbouring cells are treated using the near formula, while for the rest the far formula is used. In two dimensions this looks like this:



The interactions of the red cell with the light green cells is done via the near formula, while all the dark blue cells are treated using the far formula. One first determines the product decomposition components for each cell and then adds them up over all pairs of cells which are not neighbours.

Of course the infinite sums have to be cut off at some radius. The cutoff radii can be determined using error formulas, for example the maximal error of ϕ, calculated summing the far formula up to a pq–cutoff of R, is

$$\tau_\phi \approx \frac{e^{-2\pi R r}}{r}\,, \tag{2.26}$$

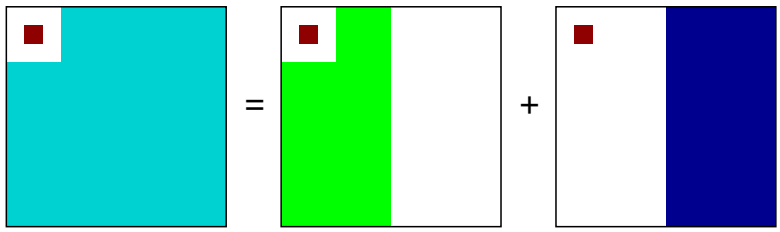
where r the distance between the particles. The error formula reflects the condition that two particles have to be separated to efficiently use the far formula. The error estimates are similar to those that will be presented for MMM2D later.

Using the algorithm described above, the minimal distance of two particles calculated with the far formula is λ_z/S. Therefore for a constant pairwise error the Fourier space cutoff R has to be chosen proportional to S. This leads to a calculation time for the far formula of $\mathcal{O}(NS^2)$. The near formula has to be used for $\mathcal{O}(N^2S^{-3})$ particle pairs. Since the calculation time for the near formula is practically parameter independent, this is also the scaling of the calculation time. The total computation time has a minimum for $S \sim N^{1/5}$, resulting in an overall computation time scaling of $\mathcal{O}(N^{\frac{7}{5}})$.

To decrease the computational effort of MMM down to $\mathcal{O}(N\log(N))$ we use the periodicity of the axis which dominates the error, i. e. in which the particles are closest. Since the far formula error of MMM2D is dominated by the error in the non–periodic direction, the following cannot be transferred to MMM2D, and i unique to the MMM method.

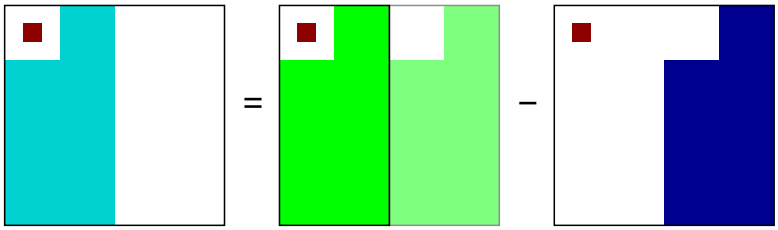
In the following we assume that the number of cells per side is a power of 2, i. e. $S = 2^L$. The main idea is to increase λ_z/S not by decreasing S, but rather by increasing λ_z. This is possible due to the periodic boundary conditions, as we will see now.

Again the idea will be presented graphically in two dimensions, and again we calculate the interactions with the small red cell. First the primary simulation cell will be divided into small cells along the x–coordinate:



The particles in the right part are far away and can be calculated using the far formula with a reasonably small cutoff.

The other half will now be calculated using a cell length of $\lambda_x/2$. This introduces artificial particles in the right part, which are just copies of the particles in the left side. Their contribution can be subtracted easily together with the contribution of the real particles in the right half.



Now we are left with a cell containing roughly $N/2$ particles (provided the simulation cell is filled homogeneously) and cell dimensions $\lambda_x/2 \times \lambda_y \times \lambda_z$. For this system we apply the same trick again to one of the other axes, e. g. y, then to z and again to x, and so on until the calculation using the near formula is more efficient than another subdivision. The subdivision occurs in the coordinate which occurs in the exponential. Therefore the shift of the particle coordinates is actually only a multiplication.

In pseudo code the algorithm for a subdivision step looks like this

for each (p, q)
for each cell c
calculate the coefficients σ_c **of the product decomposition of cell** c
end
for each row r
calculate the sum Σ_r **of all coefficients from the** σ_c **in row** r
end
for each row r
calculate the sum Φ_r **of the** Σ_r **of the rows more far away than** $\lambda_z/2$ **for row** r
add to Φ_r **the** Σ_r **of the rows closer than** $\lambda_z/2$ **but not adjacent for row** r **multiplied by the shift factor**
end
for each cell c
use the Φ_r **and the** σ_c **of the adjacent rows of all cells** c' **not adjacent to cell** c **multiplied by the shift factor to calculate the contribution to the energy from the far particles and the artificial images**
end
end

This algorithm can be further optimised with respect to computation time, but even in the present form one can see that it its implementation is more demanding than the Ewald or multipole methods. The implementation is even more complex as the presented code should work with all three coordinates symmetrically for reuse in the subdivision steps. The calculation time for each of the subdivision steps is

proportional to half of the number of particles left, i. e. $N/2, N/4, N/8, \ldots$. To maintain a constant calculation time of the near formula, one has to ensure that $2^L \sim N$ or $L \sim \log N$. Therefore the overall computation time is $\mathcal{O}(NL) = \mathcal{O}(N \log N)$.

Upper error bounds can be found easily by approximating the sums by integrals, see [92]. As for the Lekner sum, additional accuracy has to be paid with only a small decrease of computational performance, therefore MMM is the method of choice if high accuracy is required.

3 Two dimensional periodicity

Often one is interested in slab-like systems which are only periodic in two space dimensions and finite in the third, for example in problems involving electrolyte solutions between charged surfaces, proteins near charged membranes, thin films of ferrofluids, Wigner crystals, charged films, membranes, solid surfaces decorated with dipoles etc. The list of methods suitable for this case is considerably shorter:

Ewald type methods The same process leading to the Ewald method for fully periodic systems can also be used to obtain a method for two dimensional periodicity, the $2d$–Ewald sum. The resulting Fourier space sum does not allow for a product decomposition, therefore the method has a computational order of only $\mathcal{O}(N^2)$. Moreover only crude "a priori" error estimates exist, making the choice of the correct α a nontrivial task.

An alternative path in the derivation leads to the method of Hautman and Klein. This method is only suitable for small separations in the non–periodic coordinate. It is considerably faster than the $2d$–Ewald method, but also for this method no error estimates exist.

Multipole methods As said in the last chapter, multipole methods can be used for any kind of periodicity, and therefore are also usable for two dimensionally periodic systems. The efficiency is equal to the efficiency in the fully periodic case, and the methods are only competitive for more than a million particles.

Lekner sum Since the Lekner sum us obtained by transforming only one of the spatial coordinates, it can also be applied to the cases of two and one dimensional periodicity. The Lekner sum is an $\mathcal{O}(N^2)$ method, although much faster than the $2d$–Ewald method, and therefore suitable only for small numbers of particles.

MMM2D MMM can be easily adapted to the case of two dimensional periodicity. Because of the missing spatial symmetry, this method cannot achieve the $\mathcal{O}(N \log N)$ scaling of MMM, but has a computation time of $\mathcal{O}(N^{5/3})$ which is at least better than the $2d$–Ewald or Lekner methods. Like for the original MMM, the computation time grows only with the logarithm of the error bound, so that MMM2D is suitable for small numbers of particles below 1000 and high accuracy requirements. The MMM2D algorithm, error estimates and some numerical results are presented in detail in Sec. 3.3.

ELC Instead of using a specially designed method for two dimensionally periodic systems, one could replicate the system along the non–periodic coordinate. In an attempt to decouple the interactions in the third dimension, one fills only parts of the simulation box with charges and leaves some space empty. Now a standard method for three dimensional periodic systems such as P3M or MMM

can be used for the calculation of the electrostatic interaction. The size of the empty region has to be comparatively large, about $4-5$ times larger than the height of the simulation box, which slows down the methods considerably. To avoid this, the electrostatic layer correction (ELC) can be used to subtract the contribution of the image layers again. The computation time for the ELC term is only $\mathcal{O}(N)$ and therefore does not destroy the scaling of any of the known methods for three dimensional periodicity. Error estimates similar to MMM2D can be obtained. ELC will be presented in detail in chapter 4.

Method of Nijboer and de Wette Nijboer and de Wette put forward a method to calculate the interaction between particles well separated in the non–periodic dimension in computational order $\mathcal{O}(N)$ [73]. Although the derivation of Nijboer and de Wette follows a completely different line, the resulting formula is identical to the far formula of MMM2D. Since Nijboer and de Wette employ the standard spherical summation, the $2d$–Ewald sum is suitable as near formula. The combination of these methods was first reported by Smith [88]. Since the far formula is identical to MMM2D, but the near formula is less efficient, the combination method is also less efficient then MMM2D, and the sensitivity to α is another drawback, therefore MMM2D should be superior to this method in all applications.

Again, not all known methods have been presented. For example, Kawata et al recently proposed a method using yet another Ewald approach which allows a product decomposition, but at the cost of a numerical integration of different oscillating functions with a large, fixed step size [47]. This results in essentially uncontrollable errors, and no error estimates for the method exist.

In the first section of this chapter, the mathematical foundations of the Coulomb sum for two dimensionally periodic systems will be laid. The second section of this chapter will briefly show the formulas for the two Ewald type methods, the $2d$–Ewald method and the Hautman and Klein method. The third section reviews the MMM2D method. In this work the focus lies on the parallelisation of the MMM2D algorithm, which has not been addressed so far, and a detailed discussion of the computation time scaling. Using the MMM2D formulas, the ELC method will be derived in the following chapter.

3.1 General prerequisites

Without loss of generality, we assume that the simulated system is periodic in only the first two of the three spatial dimensions, namely x and y, so that the primary simulation box is

$$B_0 = \left(-\frac{\lambda_x}{2}, \frac{\lambda_x}{2}\right] \times \left(-\frac{\lambda_y}{2}, \frac{\lambda_y}{2}\right] \times \left(-\infty, \infty\right). \tag{3.1}$$

Note that since the primary simulation box contains only $N < \infty$ particles, one can nevertheless define a system height h such that $0 \leq z_i < h$ for all particles $i = 1, \ldots, N$.

The Coulomb energy of this system is given by

$$E = \frac{1}{2} \sum_{S=0}^{\infty} \sum_{k^2+l^2=S} \sum_{i,j}{}' \frac{q_i q_j}{|p_{ij} + n_{kl}|}, \tag{3.2}$$

using the notations introduced in Eqns. (2.3) through (2.7). Note that the summation again occurs in spherical order in the hyper–geometric sense. Instead of adding up the contributions of larger and larger spherical shells, here the summation occurs in the order of increasing rings. It can be shown easily, that $1/|r+n_k|+1/|r-n_k|-2/|n_k| = \mathcal{O}(|n_k|^3)$, so that for a two dimensionally periodic system the Coulomb sum the order of summation is unimportant as long as it is symmetric. Consequently, there is no shape–dependent term in the two dimensional Coulomb energy.

3.2 Ewald type methods

In this section, it is assumed that the simulation box is quadratic, i. e. $\lambda_x = \lambda_y \lambda$. Both Ewald methods do not allow a product decomposition or similar tricks, therefore the resulting formulas have to added up for all pairs of particles, with an unfavourable scaling of $\mathcal{O}(N^2)$. Moreover no error estimates exist which are as robust and accurate as the error estimates of Kolafa and Perram for the classical Ewald sum.

In the $2d$–Ewald sum, the electrostatic interaction is calculated as

$$\begin{aligned} E = \frac{1}{2} \sum_{i,j} q_i q_j \Bigg(& \sum_{n}{}' \frac{\operatorname{erfc}(\alpha |r_{ij} + \lambda n|)}{|r_{ij} + \lambda n|} + \\ & \frac{\pi}{\lambda} \sum_{m} e^{2\pi i \rho_{ij} m} F(\pi |m|, z_{ij}, \alpha) - \\ & \frac{\sqrt{\pi}}{\lambda^2 \alpha} e^{-\alpha^2 z^2} + z \operatorname{erf}(\alpha z) \Bigg), \end{aligned} \tag{3.3}$$

where $\rho_{ij} = (x_{ij}, y_{ij})$, m is two dimensional reciprocal space vector and

$$F(u, z, \alpha) = \frac{e^{2uz} \operatorname{erfc}(\frac{u}{\alpha} + \alpha z) + e^{-2uz} \operatorname{erfc}(\frac{u}{\alpha} - \alpha z)}{2u}$$

are the coefficients of the Fourier transformation along only x and y. The formula clearly resembles the classical Ewald sum in that it splits up into a real space part, a Fourier space part and a self energy contribution, but does not allow the same efficient treatment as the classic Ewald sum. The $2d$–Ewald sum is also known as the method of Heyes, Barber and Clarke or the method of de Leeuw and Perram. There exist crude error estimates [88], but they are not precise enough to tune the critical parameter α "a priori", and extensive numerical tests are necessary.

The Hautman and Klein method is only valid for small z, but converges there much faster than the $2d$–Ewald sum. The key idea is to expand the $1/r_{ij}$ interaction as

$$\frac{1}{r_{ij}} = \left(\frac{1}{r_{ij}} - \sum_{k=0}^{P} \frac{a_k z_{ij}^2 k}{\rho_{ij}^{2k+1}} \right) + \left(\sum_{k=0}^{P} \frac{a_k z_{ij}^2 k}{\rho_{ij}^{2k+1}} \right),$$

where $a_k = (-1)^k(2k)!/(2^{2k}(k!)^2)$ are the coefficients of a binomial expansion of $1/r_{ij}$ in terms of z_{ij}/ρ_{ij}, i. e. the first few terms of a Taylor expansion of $1/r_{ij}$ are extracted and treated separately. For each level of the expansion separate splitting functions are used, namely

$$\begin{aligned} h_0(s,\alpha) &= \operatorname{erf}(s/(2\alpha)) \\ h_k(s,\alpha) &= \frac{(-1)^k s^{2k+1}}{a_k(2k)!}\nabla^{2k}\frac{h_0(s,\alpha)}{s}\,. \end{aligned}$$

Using this, the electrostatic interaction is given by

$$\begin{aligned} E =& \frac{1}{2}\sum_{i,j} q_i q_j \left(\sum_n{}' \left(\frac{1}{|r_{ij}+n|} - \sum_{k=0}^{P} \frac{a_k z_{ij}^{2k} h_k(\rho_{ij},\alpha)}{|\rho_{ij}+n|^{2k+1}} \right) + \right. \\ & \left. \frac{2\pi}{\lambda^2} \sum_{k=0}^{P} a_k z_{ij}^{2k} \sum_m g_k(m,\alpha) m^{2k-1} e^{-\mathrm{i}m\rho_{ij}} \right) - \\ & \frac{1}{2\sqrt{\pi}\alpha}\sum_i q_i^2\,, \end{aligned} \tag{3.4}$$

where the second, long ranged part has been Fourier transformed, with

$$\begin{aligned} g_0(m,\alpha) &= \operatorname{erfc}(\alpha|m|) \\ g_k(m,\alpha) &= \frac{g_0(m,\alpha)}{a_k(2k)!} \end{aligned}$$

being the $2d$ Fourier transforms of the screening functions. For the method of Hautman and Klein no error estimates are known so far, but numerical tests have shown a good convergence of the method [100]. The convergence of the method only for small z is no real drawback, as it could be combined with the method of Nijboer and de Wette. Although this combination would be probably more efficient than with the $2d$–Ewald sum, it has not been reported so far.

3.3 MMM2D

The MMM2D method [4,6,7] will be repeated here briefly, since it forms the foundations for the ELC method developed in the next chapter as well as for the MMM1D method. Moreover the parallelisation of MMM2D has not been addressed so far and will be discussed in Sec. 3.3.3. Sec. 3.3.4 investigates the computation time scaling of ESPResSo more closely than the previous works, namely with respect to the number of processors in a parallel implementation and to geometrical aspects. In Sec. 3.3.5 numerical results from the (parallel) implementation of MMM2D in ESPResSo are presented.

3.3.1 The MMM2D Method

In this subsection we repeat the derivation of the formulas needed for MMM2D briefly. Similar to MMM, MMM2D uses an exponential convergence factor, i. e. the image

contribution to the potential in two dimensionally periodic boundary conditions reads

$$\tilde{\phi}_\beta(x,y,z) = \sum_{(k,l)\neq(0,0)} \frac{e^{-\beta r_{kl}}}{r_{kl}}\,, \tag{3.5}$$

while $\tilde{E}$ and ϕ_β are still defined by Eqns. (2.20) and (2.19), i. e.

$$\tilde{E} = \frac{1}{2}\lim_{\beta\to 0}\sum_{k,l,m\in\mathbb{Z}}{\sum_{i,j=1}^{N}}' \frac{q_i q_j e^{-\beta|p_{ij}+n_{klm}|}}{|p_{ij}+n_{klm}|} =: \frac{1}{2}\lim_{\beta\to 0}\sum_{i,j=1}^{N} q_i q_j \phi_\beta(x_{ij},y_{ij},z_{ij})\,,$$

where

$$\phi_\beta(x,y,z) = \tilde{\phi}_\beta(x,y,z) + \begin{cases} \frac{e^{-\beta r}}{r} & (x,y,z)\neq(0,0,0) \\ 0 & (x,y,z)=(0,0,0) \end{cases}.$$

It can be shown that $E = \tilde{E}$, i.e. the convergence factor approach gives the same result as the spherical summation, in contrast to the situation for three dimensional periodicity. For a proof of the equality of the convergence factor approach and the spherical summation order see Refs. [4,6]. The proof for the case of one dimensional periodicity given in chapter 6 is also very similar.

Transformation of ϕ_β for $z \neq 0$ — the far formula

First we concentrate on developing an absolutely and rapidly converging formula for ϕ_β. Then we can easily form the limit $\beta \to 0$ and obtain a formula for ϕ. For $z \neq 0$ and $\beta > 0$ the sum in ϕ_β is an absolutely convergent sum of Schwartz class functions. Therefore for $\delta > 0$ and $x \in \mathbb{R}$, we can apply the *Poisson formula*:

$$\sum_{k\in\mathbb{Z}} f(x+\delta k) = \frac{1}{|\delta|}\sum_{p\in\mathbb{Z}} \mathcal{F}(f)\left(\frac{p}{\delta}\right) e^{2\pi i \frac{p}{\delta}x}\,, \tag{3.6}$$

where $\mathcal{F}$ denotes the Fourier transformation. Furthermore we will be using the formulas

$$\begin{aligned} \mathcal{F}\left(\frac{e^{-\beta\sqrt{\alpha^2+\cdot^2}}}{\sqrt{\alpha^2+\cdot^2}}\right) &= 2\,\mathrm{K}_0\left(\alpha\sqrt{\beta^2+(2\pi\,\cdot)^2}\right)\,, \\ \mathcal{F}\left(\mathrm{K}_0\left(\alpha\sqrt{z^2+\cdot^2}\right)\right) &= \pi\frac{e^{-z\sqrt{\alpha^2+(2\pi\,\cdot)^2}}}{\sqrt{\alpha^2+(2\pi\,\cdot)^2}} \end{aligned} \tag{3.7}$$

which are valid for $\alpha, z \in \mathbb{R}$ and can be found, for example, in [74]. K_0 is called the modified Bessel function of order 0. For properties of the Bessel functions, see [1].

Using the same notations as for MMM (see Eqns. (3.9)) we obtain after two Fourier transformations

$$\begin{aligned} \phi_\beta(x,y,z) &= \sum_{k,l\in\mathbb{Z}} \frac{e^{-\beta r_{kl}}}{r_{kl}} = \sum_{l\in\mathbb{Z}}\left(\sum_{k\in\mathbb{Z}} \frac{e^{-\beta r_{kl}}}{r_{kl}}\right) \\ &= 2u_x \sum_{p\in\mathbb{Z}}\left(\sum_{l\in\mathbb{Z}} \mathrm{K}_0\left(\beta_p\rho_l\right)\right) e^{2\pi i u_x p x} \\ &= 2\pi u_x u_y \sum_{p,q\in\mathbb{Z}} \frac{e^{-\beta_{pq}|z|}}{\beta_{pq}} e^{2\pi i u_x p x} e^{2\pi i u_y q y}\,. \end{aligned}$$

Expanding the term for $p = q = 0$ we find $\frac{e^{-\beta|z|}}{\beta} = \beta^{-1} - |z| + \underset{\beta\to 0}{\mathcal{O}}(\beta)$, and obtain our final formula

$$\begin{aligned}\phi_\beta(x,y,z) = {} & 8\pi u_x u_y \sum_{p,q>0} \frac{e^{-\beta_{pq}|z|}}{\beta_{pq}} \cos(2\pi u_x p x)\cos(2\pi u_y q y) + \\ & 4\pi u_x u_y \sum_{q>0} \frac{e^{-\beta_q|z|}}{\beta_q} \cos(2\pi u_y q y) + 4\pi u_x u_y \sum_{p>0} \frac{e^{-\beta_p|z|}}{\beta_p} \cos(2\pi u_x p x) - \\ & 2\pi u_x u_y |z| + 2\pi u_x u_y \beta^{-1} + \underset{\beta\to 0}{\mathcal{O}}(\beta)\,,\end{aligned} \tag{3.8}$$

where

$$\begin{aligned}\beta_{pq} &= \sqrt{\beta^2 + (2\pi u_x p)^2 + (2\pi u_y q)^2}\,, \\ \beta_p = \sqrt{\beta^2 + (2\pi u_x p)^2} \quad &\text{and} \quad \beta_q = \sqrt{\beta^2 + (2\pi u_y q)^2}\,.\end{aligned} \tag{3.9}$$

It has a singularity of $2\pi u_x u_y \beta^{-1}$ which is independent of the particle coordinates. Therefore, once the sum of ϕ_β is taken over all particles, the singularity vanishes via the charge neutrality condition. For the other parts of Eqn. (3.8) taking the limit $\beta \to 0$ is trivial. The sum converges well as the summands decay exponentially, but for small z the convergence becomes poor and for $z = 0$ the sum is not defined. Thus we will need an alternative method for small z.

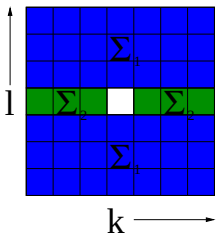
Transformation of $\tilde{\phi}_\beta$ for $z \approx 0$ — the near formula

For small particle distances, the term for $k = l = 0$ is dominant (and must be omitted for the interaction of a particle with its own images). Therefore we leave it out for now and concentrate on a rapidly convergent formula for $\tilde{\phi}_\beta(x,y,z)$. This requires a little more work. For a more detailed derivation see [92, 4].

Since we omit the $k = l = 0$ term, the area to sum over has a hole. To efficiently treat the sum over this area we split $\tilde{\phi}_\beta(x,y,z) = \Sigma_1 + \Sigma_2$ where

$$\Sigma_1 = \sum_{l\neq 0}\sum_{k\in\mathbb{Z}} \frac{e^{-\beta r_{kl}}}{r_{kl}} \quad \text{and} \quad \Sigma_2 = \sum_{k\neq 0} \frac{e^{-\beta r_{k0}}}{r_{k0}}\,.$$

Graphically this can be displayed like that:



We start by calculating Σ_1. Using the same argument as for formula (3.8) we obtain

$$\begin{aligned}\Sigma_1 &= \sum_{l\neq 0}\sum_{k\in\mathbb{Z}} \frac{e^{-\beta r_{kl}}}{r_{kl}} = 2u_x \sum_{l\neq 0}\sum_{p\in\mathbb{Z}} \mathrm{K}_0(\beta_p \rho_l) e^{2\pi i u_x p x} \\ &= 2u_x \sum_{l,p\neq 0} \mathrm{K}_0(\beta_p\rho_l) e^{2\pi i u_x p x} + 2u_x \sum_{l\neq 0} \mathrm{K}_0(\beta\rho_l)\,.\end{aligned}$$

While the first sum converges fast, the second one is still singular in β and has to be investigated further:

$$\begin{aligned} 2u_x \sum_{l\neq 0} \mathrm{K}_0(\beta\rho_l) &= 2u_x \sum_{l\in\mathbb{Z}} \mathrm{K}_0(\beta\rho_l) - 2u_x \mathrm{K}_0(\beta\rho) \\ &= 2u_x \sum_{l\in\mathbb{Z}} \mathrm{K}_0(\beta\rho_l) - 2u_x(\log 2 - \gamma - \log(\beta\rho)) + \underset{\beta\to 0}{\mathcal{O}}(\beta^2) \end{aligned}$$

where we have used the asymptotic behaviour $\mathrm{K}_0(x) = -\log\frac{x}{2} - \gamma + \mathcal{O}(x^2)$ for $x \to 0$.

The first sum can now be Fourier transformed again:

$$\begin{aligned} \sum_{l\in\mathbb{Z}} \mathrm{K}_0(\beta\rho_l) &= \pi u_y \sum_{q>0} \frac{e^{-\beta_q|z|}}{\beta_q} \left(e^{2\pi i u_y q y} + e^{-2\pi i u_y q y}\right) + \pi u_y \frac{e^{-\beta|z|}}{\beta} \\ &= 2\pi u_y \,\mathrm{Re}\left(\sum_{q>0} \frac{e^{-2\pi u_y q|z|}}{2\pi u_y q} e^{2\pi i u_y q y}\right) + \pi u_y(\beta^{-1} - |z|) + \underset{\beta\to 0}{\mathcal{O}}(\beta)\,. \end{aligned}$$

If we set $\zeta := 2\pi u_y(|z| + iy)$, we obtain

$$2\pi u_y \sum_{q>0} \frac{e^{-2\pi u_y q|z|}}{2\pi u_y q} e^{2\pi i u_y q y} = \sum_{q>0} \frac{e^{-q\zeta}}{q} = -\log(\zeta) + \frac{\zeta}{2} - \sum_{n\geq 2} \frac{b_n}{n\,n!}\zeta^n$$

where b_n are the Bernoulli numbers. This series expansion is valid only for $|\zeta| \leq 2\pi$, which is fulfilled if $|z| \leq \frac{\lambda_y}{2}$. The last equality is found by integration from

$$\begin{aligned} \frac{d}{dz}\log\left(\frac{1-e^{-z}}{z}\right) &= z^{-1}\left(\frac{z}{e^z-1} - 1\right) \\ &= z^{-1}\left(\sum_{n=0}^{\infty} \frac{b_n}{n!} z^n - 1\right) = -\frac{1}{2} + \sum_{n=1}^{\infty} \frac{b_{2n}}{(2n)!} z^{2n-1}\,, \end{aligned}$$

where the defining series $\frac{z}{e^z-1} = \sum_{n=0}^{\infty} \frac{b_n}{n!} z^n$ for the Bernoulli numbers was used.

Using $\mathrm{Re}(-\log\zeta) = -\log|\zeta| = -\log(2\pi u_y \rho)$ and $\mathrm{Re}\left(\frac{\zeta}{2}\right) = \pi u_y |z|$ we obtain

$$\sum_{l\in\mathbb{Z}} \mathrm{K}_0(\beta\rho_l) = -\sum_{n\geq 2} \frac{b_n}{n\,n!} \mathrm{Re}\,(\zeta^n) - \log(2\pi u_y \rho) + \pi u_y \beta^{-1} + \underset{\beta\to 0}{\mathcal{O}}(\beta)\,.$$

It is easy to see that ζ can be replaced by $\xi := 2\pi u_y(z + iy)$ without changing the value of the sum. This is of advantage for the calculation of the forces by differentiation.

Combining everything we obtain

$$\begin{aligned} \Sigma_1 = {} & 2u_x \sum_{l,p\neq 0} \mathrm{K}_0(\beta_p\rho_l) e^{2\pi i u_x p x} - 2u_x \sum_{n\geq 2} \frac{b_n}{n\,n!} \mathrm{Re}\big((2\pi u_y(z+iy))^n\big) - \\ & 2u_x \log(4\pi u_y) + 2u_x \log(\beta) + 2\pi u_x u_y \beta^{-1} + 2u_x\gamma + \underset{\beta\to 0}{\mathcal{O}}(\beta)\,. \end{aligned} \tag{3.10}$$

Now we concentrate on $\Sigma_2 = \sum_{k>0} \frac{e^{-\beta r_k}}{r_k} + \sum_{k<0} \frac{e^{-\beta r_k}}{r_k}$. It is sufficient to investigate the first of the two sums, because by replacing x by $-x$ the value of the second sum can be obtained.

We start with

$$\begin{aligned}\sum_{k>0} \frac{e^{-\beta r_k}}{r_k} &= \sum_{k>0} e^{-\beta r_k} \left(\frac{1}{r_k} - \frac{1}{k\lambda_x}\right) + \sum_{k>0} \frac{e^{-\beta r_k}}{k\lambda_x} \\ &= \sum_{k>0} \left(\frac{1}{r_k} - \frac{1}{k\lambda_x}\right) + \underset{\beta\to 0}{\mathcal{O}}(\beta) + \sum_{k>0} \frac{e^{-\beta r_k}}{k\lambda_x}\,.\end{aligned}$$

Details about the precise derivation of the equation can again be found in [92, 4].

Moreover by $r_k - k\lambda_x = \underset{k\to\infty}{\mathcal{O}}(1)$ and $\log(1 - e^{-\alpha}) = \log\alpha + \mathcal{O}(\alpha)$ we obtain

$$\sum_{k>0} \frac{e^{-\beta r_k}}{k\lambda_x} = \sum_{k>0} \frac{e^{-\beta k\lambda_x}}{k\lambda_x} \left(1 + \underset{\beta\to 0}{\mathcal{O}}(\beta)\right) = -u_x \log(\lambda_x \beta) + \underset{\beta\to 0}{\mathcal{O}}(\beta\log\beta)\,.$$

To evaluate the sum $\sum_{k>0} \left(\frac{1}{r_k} - \frac{1}{k\lambda_x}\right)$, we consider a N_ψ such that $N_\psi > u_x\rho + 1$. Then for $k \geq N_\psi$ we have $\rho/|x + k\lambda_x| \leq 1$ and therefore we can use the binomial series for $\sqrt{1 + \frac{\rho^2}{|x+k\lambda_x|^2}}$ to obtain

$$\begin{aligned}&\sum_{k\geq N_\psi} \left(\frac{1}{r_k} - \frac{1}{k\lambda_x}\right) \\ &\quad= \sum_{k\geq N_\psi} \left(\frac{1}{|x + k\lambda_x|} - \frac{1}{k\lambda_x}\right) + \sum_{n>0} \binom{-\frac{1}{2}}{n} \sum_{k\geq N_\psi} \frac{\rho^{2n}}{|x + k\lambda_x|^{2n+1}} \\ &\quad= -u_x\psi^{(0)}(N_\psi + u_x x) - u_x\gamma - u_x \sum_{n>0} \binom{-\frac{1}{2}}{n} \frac{\psi^{(2n)}(N_\psi + u_x x)}{\lambda_x^{2n}(2n)!} (u_x\rho)^{2n}\end{aligned}$$

where $\psi^{(n)}$ are the polygamma functions. For details on these functions, see [1].

In summary we obtain for Σ_2:

$$\begin{aligned}\Sigma_2 = -u_x \sum_{n\geq 0} \binom{-\frac{1}{2}}{n} \frac{\left(\psi^{(2n)}(N_\psi + u_x x) + \psi^{(2n)}(N_\psi - u_x x)\right)}{(2n)!} (u_x\rho)^{2n} - \\ 2u_x\gamma + 2u_x\log(u_x) - 2u_x\log(\beta) + \sum_{k=1}^{N_\psi - 1} \left(\frac{1}{r_k} + \frac{1}{r_{-k}}\right) + \underset{\beta\to 0}{\mathcal{O}}(\beta\log\beta)\,.\end{aligned}$$

Combining the formulas for Σ_1 and Σ_2 we obtain, for $|x| \leq \frac{\lambda_x}{2}$, $|y|, |z| \leq \frac{\lambda_y}{2}$ and N_ψ

such that $N_\psi > u_x\rho + 1$, the final formula

$$\begin{aligned}\tilde{\phi}_\beta(x,y,z) = &4u_x \sum_{l,p>0} \left(\mathrm{K}_0(\beta_p\rho_l) + \mathrm{K}_0(\beta_p\rho_{-l})\right)\cos(2\pi u_x p x) - \\ &2u_x \sum_{n\geq 1} \frac{b_{2n}}{2n(2n)!}\,\mathrm{Re}\left((2\pi u_y(z+iy))^{2n}\right) + \sum_{k=1}^{N_\psi-1}\left(\frac{1}{r_k}+\frac{1}{r_{-k}}\right) - \\ &u_x \sum_{n\geq 0}\binom{-\frac{1}{2}}{n}\frac{\left(\psi^{(2n)}(N_\psi+u_x x)+\psi^{(2n)}(N_\psi-u_x x)\right)}{(2n)!}(u_x\rho)^{2n} - \\ &2u_x\log(4\pi\frac{u_y}{u_x}) + 2\pi u_x u_y\beta^{-1} + \underset{\beta\to 0}{\mathcal{O}}(\beta\log\beta)\,.\end{aligned} \tag{3.11}$$

Of course this formula leads to the same singularity in β as formula (3.8) and the charge neutrality argument also holds for any combination of the two formulas as long as the sum is performed over all particles.

Energy expressions

For the implementation the formulas after taking the limit $\beta \to 0$ are needed, which will be given now. For $z \neq 0$ we obtain the *far formula in two dimensions*

$$\begin{aligned}\phi(x,y,z) = \; &4u_x u_y \sum_{p,q>0}\frac{e^{-2\pi f_{pq}|z|}}{f_{pq}}\cos(\omega_p x)\cos(\omega_q y) + \\ &2u_x u_y\left(\sum_{q>0}\frac{e^{-2\pi f_q|z|}}{f_q}\cos(\omega_q y) + \sum_{p>0}\frac{e^{-2\pi f_p|z|}}{f_p}\cos(\omega_p x)\right) - 2\pi u_x u_y|z|\,,\end{aligned} \tag{3.12}$$

where f_{pq} etc. are defined as for MMM by Eqn.(2.23).

For $|x| \leq \frac{\lambda_x}{2}$, $|y|, |z| \leq \frac{\lambda_y}{2}$ and N_ψ such that $N_\psi > u_x\rho + 1$, the *near formula in two dimensions* is given by

$$\begin{aligned}\tilde{\phi}(x,y,z) = &4u_x \sum_{l,p>0}\left(\mathrm{K}_0(\omega_p\rho_l) + \mathrm{K}_0(\omega_p\rho_{-l})\right)\cos(\omega_p x) - \\ &2u_x \sum_{n\geq 1}\frac{b_{2n}}{2n(2n)!}\,\mathrm{Re}\left((2\pi u_y(z+iy))^{2n}\right) + \sum_{k=1}^{N_\psi-1}\left(\frac{1}{r_k}+\frac{1}{r_{-k}}\right) - \\ &u_x \sum_{n\geq 0}\binom{-\frac{1}{2}}{n}\frac{\left(\psi^{(2n)}(N_\psi+u_x x)+\psi^{(2n)}(N_\psi-u_x x)\right)}{(2n)!}(u_x\rho)^{2n} - \\ &2u_x\log\left(4\pi\frac{u_y}{u_x}\right)\,.\end{aligned} \tag{3.13}$$

Finally the *self–energy in two dimensions* is

$$\tilde{\phi}(0,0,0) = 8u_x \sum_{l,p>0}\mathrm{K}_0(2\pi u_x\lambda_y p l) + 2u_x\psi^{(0)}(1) - 2u_x\log\left(4\pi\frac{u_y}{u_x}\right)\,. \tag{3.14}$$

As said before, the far formula was derived previously in a completely different way for a conventional spherical limit by Nijboer and de Wette [88,73], which requires an Ewald method for two dimensional periodicity as counterpart. The evaluation of the near formula (3.13) is significantly faster and rigorous error estimates exist, therefore MMM2D should be faster than any method using an Ewald type near formula.

Force expressions

Since the sums in equations (3.12) rsp. (3.13) converge absolutely, the electrostatic force $F_i = -\nabla_{p_i} E$ can be derived by simple term-wise differentiation and the force can be calculated as

$$F_i = \sum_{j=1}^{N} q_i q_j F(x_i - x_j, y_i - y_j, z_i - z_j)$$

where for $z \neq 0$ we obtain $F = (F_x, F_y, F_z)^T$ where

$$\begin{aligned} F_x(x,y,z) = & 8\pi u_x^2 u_y \sum_{p,q>0} p \frac{e^{-2\pi f_{pq}|z|}}{f_{pq}} \sin(\omega_p x)\cos(\omega_q y) + \\ & 4\pi u_x u_y \sum_{p>0} e^{-2\pi f_p |z|} \sin(\omega_p x)\,, \end{aligned}$$

$$\begin{aligned} F_y(x,y,z) = & 8\pi u_x u_y^2 \sum_{p,q>0} q \frac{e^{-2\pi f_{pq}|z|}}{f_{pq}} \cos(\omega_p x)\sin(\omega_q y) + \\ & 4\pi u_x u_y \sum_{q>0} e^{-2\pi f_q |z|} \sin(\omega_q y) \end{aligned} \tag{3.15}$$

and

$$\begin{aligned} F_z(x,y,z) = & 8\pi \operatorname{sign}(z) u_x u_y \sum_{p,q>0} e^{-2\pi f_{pq}|z|} \cos(\omega_p x)\cos(\omega_q y) + \\ & 4\pi \operatorname{sign}(z) u_x u_y \sum_{q>0} e^{-2\pi f_q |z|} \cos(\omega_q y) + \\ & 4\pi \operatorname{sign}(z) u_x u_y \sum_{p>0} e^{-2\pi f_p |z|} \cos(\omega_p x) \; + 2\pi \operatorname{sign}(z) u_x u_y\,. \end{aligned}$$

For $|x| \leq \frac{\lambda_x}{2}$, $|y|, |z| \leq \frac{\lambda_y}{2}$ and N_ψ such that $N_\psi > u_x \rho + 1$, we have

$$F(x,y,z) = \tilde{F}(x,y,z) + \begin{cases} \frac{1}{(x^2+y^2+z^2)^{\frac{3}{2}}} (x,y,z)^T & (x,y,z) \neq (0,0,0) \\ 0 & (x,y,z) = (0,0,0) \end{cases} \tag{3.16}$$

where

$$\begin{aligned}\tilde{F}_x(x,y,z) = &8\pi u_x^2 \sum_{l,p>0} p\left(\mathrm{K}_0(\omega_p\rho_l) + \mathrm{K}_0(\omega_p\rho_{-l})\right)\sin(\omega_p x) + \\ &\sum_{k=1}^{N_\psi-1}\left(\frac{x+k\lambda_x}{r_k^3} + \frac{x-k\lambda_x}{r_{-k}^3}\right) + \\ &u_x^2\sum_{n\geq 0}\binom{-\frac{1}{2}}{n}\frac{\left(\psi^{(2n+1)}(N_\psi+u_x x) - \psi^{(2n+1)}(N_\psi-u_x x)\right)}{(2n)!}(u_x\rho)^{2n}\,,\end{aligned}$$

$$\begin{aligned}\tilde{F}_y(x,y,z) = &8\pi u_x^2 \sum_{l,p>0} p\left(\frac{(y+l\lambda_y)\,\mathrm{K}_1(\omega_p\rho_l)}{\rho_l} + \frac{(y-l\lambda_y)\,\mathrm{K}_1(\omega_p\rho_{-l})}{\rho_{-l}}\right)\cos(\omega_p x) - \\ &4\pi u_y u_x \sum_{n\geq 1}\frac{b_{2n}}{(2n)!}\,\mathrm{Im}\big((2\pi u_y(z+iy))^{2n-1}\big) + \sum_{k=1}^{N_\psi-1}\left(\frac{y}{r_k^3} + \frac{y}{r_{-k}^3}\right) + \\ &u_x^3 y\sum_{n\geq 1}\binom{-\frac{1}{2}}{n}\frac{\left(\psi^{(2n)}(N_\psi+u_x x) + \psi^{(2n)}(N_\psi-u_x x)\right)}{(2n-1)!}(u_x\rho)^{2(n-1)}\end{aligned}$$

and

$$\begin{aligned}\tilde{F}_z(x,y,z) = &8\pi u_x^2 \sum_{l,p>0} p\left(\frac{z\,\mathrm{K}_1(\omega_p\rho_l)}{\rho_l} + \frac{z\,\mathrm{K}_1(\omega_p\rho_{-l})}{\rho_{-l}}\right)\cos(\omega_p x) - \\ &4\pi u_y u_x \sum_{n\geq 1}\frac{b_{2n}}{(2n)!}\,\mathrm{Re}\big((2\pi u_y(z+iy))^{2n-1}\big) + \sum_{k=1}^{N_\psi-1}\left(\frac{z}{r_k^3} + \frac{z}{r_{-k}^3}\right) + \\ &u_x^3 z\sum_{n\geq 1}\binom{-\frac{1}{2}}{n}\frac{\left(\psi^{(2n)}(N_\psi+u_x x) + \psi^{(2n)}(N_\psi-u_x x)\right)}{(2n-1)!}(u_x\rho)^{2(n-1)}\,.\end{aligned}$$

Here K_1 is the modified Bessel function of order one.

3.3.2 Error Estimates

For an implementation which is of practical use, error estimates are needed. Since we calculate the energy (forces) by summing up pairwise contributions $q_i q_j \phi(x_{ij}, y_{ij}, z_{ij})$ (their differential), it is reasonable to derive an upper bound for the error of ϕ. The maximal pairwise error of the pair energy is induced by the calculation of the formulas (3.12) and (3.13) with finite cutoffs. Likewise we can derive an maximal pairwise error ϵ_F for the pair forces by formulas (3.15) and (3.16). From this one can give an upper bound for the commonly used RMS error of the forces. We will show later that the error distribution for MMM2D is highly non–uniform, and leads typically to an RMS error that is much lower than our bound. Thus the RMS error is not the optimal error measure. Furthermore our implementation shows that an increase in precision has little impact on the calculation time of MMM2D, quite contrary to mesh based [19] and other methods [100].

For the far formula given by (3.12) and (3.15), respectively, we use a radial cutoff. So the summation is not performed over all $(p,q) \neq 0$, but only for those $(p,q) \in \Gamma_R$

where

$$\begin{aligned}\Gamma_R = &\left\{(p,q) \in \mathbb{Z}^2 \,|\, p,q > 0, u_x^2(p-1)^2 + u_y^2(q-1)^2 < R^2\right\} \cup \\ &\{(p,0) \in \mathbb{Z} \times \{0\} \,|\, u_x p < R\} \\ &\{(0,q) \in \{0\} \times \mathbb{Z} \,|\, u_y p < R\}\ .\end{aligned} \tag{3.17}$$

The three sets correspond to the three sums over the (p,q)–vectors. The somewhat complicated form of Γ_R is necessary to obtain a strict, but sharp upper bound. By a simple approximation of the sums by integrals it is easy to derive upper bounds on the remainder of the sum. For the potential ϕ we find the estimate

$$|\phi(x,y,z) - \phi(R,x,y,z)| \le \tau_\phi^{far} := \left(1 + \frac{u_x + u_y}{\pi R}\right) \frac{e^{-2\pi R|z|}}{|z|}\,. \tag{3.18}$$

This upper bound is *not* valid for the forces. For all three components F_x, F_y and F_z of the forces we find the upper bound

$$|F(x,y,z) - F(R,x,y,z)| \le \tau_F^{far} = \frac{e^{-2\pi R|z|}}{|z|}\left(2\pi R + 2(u_x+u_y) + \frac{1}{|z|}\right)\,. \tag{3.19}$$

This value is precise only for F_z. The other force components show a better convergence. Note that $\tau_\phi \le \tau_F$, so that the potential is at least as precise as the force components. Common to both error formulas is the exponential decrease of the error with $R|z|$.

The near formula given by equations (3.13) rsp. (3.16) contains three sums with different cutoffs. For the first sum containing Bessel functions it is reasonable to sum over all $(p,q) \in \Omega_L$ where

$$\Omega_L := \left\{(p,l) \;\middle|\; 0 < p < \frac{L}{\pi u_x} \quad \text{and} \quad 0 < l < \frac{L}{\omega_p} + 1 \right\}\,. \tag{3.20}$$

Again the form is dictated by the requirements of a strict, but sharp upper bound. The Bessel functions are replaced according to $\mathrm{K}_0(u) < \mathrm{K}_1(u) < \mathrm{K}_1(u_0)e^{u_0}e^{-u}$ for all $u \ge u_0$. Let $L \ge \pi u_x + u_y$, $|x| \le \lambda_x/2$, $|y|, |z| \le \lambda_y/2$, then the upper bound

$$\tau_E^{Bessel} = 8u_x\,\mathrm{K}_0(\lambda_y L)\left(\frac{e^{\pi u_x \lambda_y}}{\pi u_x \lambda_y} + \sum_{p=1}^{\lceil \frac{L}{\pi u_x}\rceil - 1} p e^{-\pi u_x \lambda_y p}\right) \tag{3.21}$$

for the potential can be found again by an approximation of the sum by an integrals. The prefactor $\mathrm{K}_1(\lambda_y L)e^{\lambda_y L}$ is smaller than 1 for $L \ge 3u_y$. Similarly the upper bound

$$\tau_F^{Bessel} = 16\pi u_x^2\,\mathrm{K}_1(\lambda_y L)\left(\frac{e^{\pi u_x \lambda_y}}{\pi u_x \lambda_y}\left(\frac{L+u_y}{\pi u_x} - 1\right) + \sum_{p=1}^{\lceil \frac{L}{\pi u_x}\rceil - 1} p e^{-\pi u_x \lambda_y p}\right)$$

for all force components can be obtained. Again $\max(1, 1/(2\pi u_x))\tau_F^{Bessel}$ is a uniform upper limit for both the potential and the force components. Note that K_1 drops faster than exponentially and therefore also both error limits. The Gaussian bracket $\lceil\cdot\rceil$ denotes rounding up.

For the second sum in (3.13) containing the Bernoulli numbers we use the estimate $|b_{2n}| \leq 4\frac{(2n)!}{(2\pi)^{2n}}$ [1] to obtain the error estimates

$$\tau_E^{Bernoulli} = 16u_x(u_y\rho)^{2N} \leq 16u_x 2^{-N} \tag{3.22}$$

and

$$\tau_F^{Bernoulli} = 16u_x u_y(u_y\rho)^{2N-1} \leq 16\sqrt{2}u_x u_y 2^{-N} \tag{3.23}$$

for the summation of this sum up to N. Note that this sum does not contribute to F_x, due to the artificially broken symmetry in the derivation. The error estimate contains the particle position dependent term ρ. By using a table lookup scheme for $N(\rho)$ one can chose the appropriate cutoff at runtime to speed up the calculation.

For the last sum containing the polygamma functions there is no error estimate necessary, as it is a Leibniz sum. The cutoff is determined at runtime by adding up terms until they are smaller than a threshold $\tau_{E,F}^{\psi}$. The overall maximal error of the near formula is then $\tau_{E,F}^{near} = \tau_{E,F}^{Bessel} + \tau_{E,F}^{Bernoulli} + \tau_{E,F}^{\psi}$.

Note that although the use of the RMS force error may be misleading as the error distribution is non–homogenous, it is still possible to obtain a weak upper bound on the RMS force error. Following the arguments presented in [20], one can see that the RMS force error is given by

$$\Delta F_{RMS} = \frac{\sum_{i=1}^{N} q_i^2}{\sqrt{N}}\chi\,, \tag{3.24}$$

where χ is assumed to be the RMS force error introduced by the algorithm to the pair interaction of two particles. For Eqn. (3.24) to hold one assumes a homogenous particle distribution and that the particle pair errors are statistically independent. Obviously in the case of MMM2D $\chi \leq \sqrt{3}\tau_F$, where $\tau_F = \max(\tau_F^{far}, \tau_F^{near})$, which then gives

$$\Delta F_{RMS} \leq \sqrt{\frac{3}{N}}\sum_{i=1}^{N} q_i^2\tau_F\,. \tag{3.25}$$

This means that for a system of 500 unit charges and 500 negative unit charges $\Delta F_{RMS} \leq 54.78\tau_F$, so that tuning the pairwise error to 10^{-4} results in a RMS force error less than $5 \cdot 10^{-3}$, which is sufficient for most applications.

3.3.3 Parallel Implementation

In [4,6] one can find an algorithm for implementing Eqns. (3.12) and (3.13) efficiently such that the computation time scales like $\mathcal{O}(N^{\frac{5}{3}})$. However, the parallelisation of the MMM2D algorithm has not been addressed so far, and will be discussed here, as well as the integration into a standard MD integration scheme.

Because of the broken symmetry of the two dimensional periodic problem the two periodic coordinates needed for the far formula are necessarily x and y, and particles with a similar z–coordinate have to be treated using the near formula, even if they are well separated in x or y. Therefore instead of dividing the simulation box into cells in all spatial dimensions as for the standard linked–cell structure, we divide it into B equally sized layers only along the z–axis, of which each has a height of $b = h/B$. h is an artificial box height large enough to contain all the particles.

We assume that a layer will be treated by only one processor, although a processor might deal with several layers. In general the optimal number of layers is larger than eight, therefore using two processors per layer requires at least 16 processors, which is normally unacceptable for systems with less than 10000 particles.

The processors are basically stacked up along the z–axis, so that they have a spatial ordering. We assume that the n_{proc} processors are numbered according to the z–coordinate of their dedicated layers, so that processor 1 is the bottom processor and processor n_{proc} the top processor. As already described for MMM we use the near formula for adjacent layers, while for the rest the far formula is used.

Treatment of the near formula

In the following we assume that for the short ranged interactions a link cell algorithm (see Sec. 1.3) is used, where the cells are the layers described here, e. g. the layered cell system of ESPResSo. In this case, the maximal distance of two particles in adjacent layers is $d_{near} = 2b+r_{skin}$, while the minimal distance of two particles in non–adjacent layers is $d_{far} = b-r_{skin}$. The near formula interaction can be treated as a short ranged interaction in which case b must be larger or equal to the maximal interaction range of the other short ranged interactions. Moreover for the the near formula to be valid we need $|z_i - z_j| \leq \lambda_y/2$ for particles i and j located in adjacent layers. This gives the constraint

$$2b = \frac{2h}{B} \leq \frac{\lambda_y}{2}\,. \tag{3.26}$$

Therefore λ_y should be as large as possible and it might be necessary to exchange the x– and y–axis if $\lambda_x >> \lambda_y$. If the maximal interaction range of any other short ranged interaction is larger than $1/4\max(\lambda_x, \lambda_y)$, the treatment of the short–ranged interactions and the treatment of the near formula have to be decoupled. Since the near formula has to added up over all particles in the adjacent layers, a Verlet list is of little use in combination with MMM2D.

The self energy $\sum_{i=1}^{N} q_i^2 \tilde{\phi}(0,0,0)$ should be added separately, as normally the sum over all charges squares does not change.

Treatment of the far formula

The calculation of the far formula consists of summing up terms with frequencies (p, q) in the Fourier space. By virtue of a product decomposition this can be done such that the computing time is $\mathcal{O}(N)$. The algorithm presented can easily be adapted for the calculation of the sums with only a single Fourier frequency and the $|z|$–sum. The same holds for the calculation of the forces.

In the beginning we concentrate on a single particle i located in layer S_i. The far formula is used to calculate the contributions from all particles in non–adjacent layers, i. e. layers S_j where $|S_j - S_i| > 1$. First we restrict attention to the particles j in layers $S_j < S_i - 1$, that is, to the particles in the set $\bigcup_{S<S_i-1} I_S$. The cosine terms are separable via the addition theorem and since $|z_i - z_j| = z_i - z_j$ for these layers,

we have

$$\begin{aligned}
&\sum_{\substack{j\in I_S\\ S<S_i-1}} q_iq_j\frac{e^{-2\pi f_{pq}|z_i-z_j|}}{f_{pq}}\cos(\omega_p(x_i-x_j))\cos(\omega_q(y_i-y_j)) = \\
&q_i\frac{e^{-2\pi f_{pq}z_i}}{f_{pq}}\cos(\omega_p x_i)\cos(\omega_q y_i)\sum_{\substack{j\in I_S\\ S<S_i-1}} q_j e^{2\pi f_{pq}z_j}\cos(\omega_p x_j)\cos(\omega_q y_j)+ \\
&q_i\frac{e^{-2\pi f_{pq}z_i}}{f_{pq}}\cos(\omega_p x_i)\sin(\omega_q y_i)\sum_{\substack{j\in I_S\\ S<S_i-1}} q_j e^{2\pi f_{pq}z_j}\cos(\omega_p x_j)\sin(\omega_q y_j)+ \\
&q_i\frac{e^{-2\pi f_{pq}z_i}}{f_{pq}}\sin(\omega_p x_i)\cos(\omega_q y_i)\sum_{\substack{j\in I_S\\ S<S_i-1}} q_j e^{2\pi f_{pq}z_j}\sin(\omega_p x_j)\cos(\omega_q y_j)+ \\
&q_i\frac{e^{-2\pi f_{pq}z_i}}{f_{pq}}\sin(\omega_p x_i)\sin(\omega_q y_i)\sum_{\substack{j\in I_S\\ S<S_i-1}} q_j e^{2\pi f_{pq}z_j}\sin(\omega_p x_j)\sin(\omega_q y_j)\,.
\end{aligned} \tag{3.27}$$

The sum for the $S > S_i+1$ is very similar, just the sign of the z_i and z_j is exchanged. For all particles j only the eight terms

$$\xi_j^{(\pm,s/c,s/c)} = q_j e^{\pm 2\pi f_{pq}z_j}\sin/\cos(\omega_p x_j)\sin/\cos(\omega_q y_j) \tag{3.28}$$

are needed. The upper index describes the sign of the exponential term and whether sine or cosine is used for x_j and y_j in the obvious way. These terms can be used for all expressions on the right hand side of Eqn. (3.27). Moreover it is easy to see from the addition theorem for the sine function that these terms also can be used to calculate the force information up to simple prefactors that depend only on p and q.

Every processor starts with the calculation of the terms $\xi_j^{(\pm,s/c,s/c)}$ and adds them up in each layer, so that one obtains

$$\Xi_s^{(\pm,s/c,s/c)} = \sum_{j\in S_s}\xi_j^{(\pm,s/c,s/c)}\,. \tag{3.29}$$

Now we calculate

$$\begin{aligned}
\Xi_s^{(l,s/c,s/c)} &= \sum_{t<s-1}\Xi_t^{(+,s/c,s/c)} \quad \text{and} \\
\Xi_s^{(h,s/c,s/c)} &= \sum_{t>s+1}\Xi_t^{(-,s/c,s/c)}\,,
\end{aligned} \tag{3.30}$$

which are needed for the evaluation of Eqn. (3.27). While the bottom processor can calculate $\Xi_s^{(l,s/c,s/c)}$ directly, the other processors are dependent on its results. Therefore the bottom processor starts with the calculation of its $\Xi_s^{(l,s/c,s/c)}$ and sends up $\Xi_s^{(l,s/c,s/c)}$ and $\Xi_s^{(+,s/c,s/c)}$ of its top layer s to the next processor dealing with the layers above. Simultaneously the top processor starts with the calculation of the $\Xi_s^{(h,s/c,s/c)}$ and sends them down. In total $n_{proc}-1$ communication rounds with two

independent communications are needed. Now every processor can use the $\Xi_j^{(l/h,s/c,s/c)}$ and the $\xi_j^{(\pm,s/c,s/c)}$ to calculate the force rsp. energy contributions for its particles.

One should make sure that the Fourier cutoff R is not too large. Otherwise the $\xi_j^{(\pm,s/c,s/c)}$–terms might get too large rsp. too small to be represented by a regular double precision floating point number. For a 16–bit exponent the limit is above $R = 100u_z$. This is uncritical since even for a few hundred particles $R > 100u_z$ results in an unacceptable computation time.

Finally here is the complete algorithm for the calculation of the contribution of one (p, q)–vector of the far formula on a single node in pseudo code:

for each layer $s = 1 \dots S$
$\Xi_s^{(\pm,s/c,s/c)} = 0$
for each particle j **in layer** s
calculate $\xi_j^{(\pm,s/c,s/c)}$
$\Xi_s^{(\pm,s/c,s/c)} += \xi_j^{(\pm,s/c,s/c)}$
end
end
$\Xi_3^{(l,s/c,s/c)} = \Xi_1^{(+,s/c,s/c)}$
for each layer $s = 4 \dots S$
$\Xi_s^{(l,s/c,s/c)} = \Xi_{s-1}^{(l,s/c,s/c)} + \Xi_{s-2}^{(+,s/c,s/c)}$
end
$\Xi_{S-2}^{(l,s/c,s/c)} = \Xi_S^{(-,s/c,s/c)}$
for each layer $s = (S-3) \dots 1$
$\Xi_s^{(l,s/c,s/c)} = \Xi_{s+1}^{(l,s/c,s/c)} + \Xi_{s+2}^{(-,s/c,s/c)}$
end
for each layer $s = 1 \dots S$
for each particle j **in layer** s
calculate particle interaction from $\xi_j^{(+,s/c,s/c)}\Xi_s^{(l,s/c,s/c)}$ **and** $\xi_j^{(-,s/c,s/c)}\Xi_s^{(h,s/c,s/c)}$
end
end

For multiple processor nodes only the two middle loops have to be modified such that the sums are transferred across the processor boundaries. Because the index of the added $\Xi_t^{(\pm,s/c,s/c)}$ and of the $\Xi_s^{(l/h,s/c,s/c)}$ differ by two, two of the $\Xi_s^{(l/h,s/c,s/c)}$ depend on data from the neighbour node, so that one has to transfer both the current sum and the last summand or something similar.

3.3.4 Efficiency

The same calculation as was done for the computation time of MMM can be done for MMM2D. We assume that N/B particles are located in every layer, e. g. a homogenous particle distribution. The calculation of the $\xi_j^{(\pm,s/c,s/c)}$, the $\Xi_j^{(\pm,s/c,s/c)}$ and the final contributions of the far formula use a computation time of $\mathcal{O}(\lambda_x\lambda_y R^2 N/n_{proc})$. The calculation of the $\Xi_j^{(l/h,s/c,s/c)}$ consumes a computation time of $\mathcal{O}(\lambda_x\lambda_y R^2 B)$. The prefactor $\lambda_x\lambda_y R^2$ accounts for the number of (p,q) vectors used.

The near formula needs a computation time of $\mathcal{O}(T_n N^2/B/n_{proc})$, where T_n is the computation time of a single evaluation of the near formula. Unfortunately T_n shows a very complex behaviour with respect to the B and the dimensions. For example, the computation time of the Bernoulli sum depends on the fraction $u_y h/B$, since this fraction determines the maximal size of $u_y\rho \leq u_y/2\sqrt{h^2+\lambda_y^2}$. The convergence speed of the polygamma sum depends on $u_x\rho$, and if this value is large, a considerably large number of terms have to be added until the exponential convergence breaks through. Note that these are all logarithmic dependencies, so as long as $u_y h/B$ is not close to the limit $B \geq 4u_y h$ (see Eqn. (3.26)), the computation time of the near formula is practically constant. The calculation of the self–energy needs a computation time of $\mathcal{O}(N/n_{proc}\log\epsilon^{-1})$, if one neglects the calculation of the single particle self–energy, which is done only once.

The minimal distance of two particles that are treated by the far formula is $b = h/B$, omitting an eventually skin from the link–cell algorithm. From the error estimates (3.18) we obtain for the potential using the far formula a scaling of

$$R \sim \frac{B}{h}\log\left(\frac{B}{\epsilon\, h}\right). \tag{3.31}$$

For the force the log term has a more complex dependence on λ_x, λ_y and h, which cannot be given explicitly, as the error estimate is not analytically invertible. But because for real applications the error bound ϵ is much smaller than the other factors contained in the log term, namely B and h, the term is practically constant for a real system both for the force and the potential evaluation. Therefore it will be neglected in the following.

Assuming a nearly constant computation time for the near formula, i. e. a not too small number of layers B, the box dimensions enter the computation time primarily through the number of (p,q) vectors $\pi\lambda_x\lambda_y R^2$ as a prefactor

$$\theta = \lambda_x\lambda_y/h^2\,. \tag{3.32}$$

We obtain the full asymptotic computation time as

$$\mathcal{O}\left(\frac{\theta N B^2}{n_{proc}}\right) + \mathcal{O}\left(\theta B^3\right) + \mathcal{O}\left(\frac{N^2}{B n_{proc}}\right) + \mathcal{O}\left(\frac{N}{n_{proc}}\right), \tag{3.33}$$

where the first term accounts for the calculation of the $\xi_j^{(\pm,s/c,s/c)}$ and the final contributions of the far formula, the second term accounts for the calculation of the $\Xi_j^{(l/h,s/c,s/c)}$ including the communication, the third term accounts for the near formula and the last term accounts for the calculation of the self energy. Note that

the second term $\mathcal{O}(\theta B^3)$ is much smaller than the other terms since the number of layers should be much smaller than the number of particles. Consequently, this term is ignored in the following.

The scaling of a method is normally given for a constant number of processors and box dimensions. In this case Eqn. (3.33) is minimised by

$$B_{ideal}(N) = \mathcal{O}\left(N^{1/3}\right) , \tag{3.34}$$

yielding the optimal computation time

$$T_{ideal}(N) = \mathcal{O}\left(N^{5/3}\right) . \tag{3.35}$$

As B/h scales like $N^{1/3}$, it will be small compared to ϵ. In our implementation for 10000 particles and a box length of $h = 10$ we find the optimal $B \sim 30$, so that we can safely ignore the additional log term at reasonable precision settings. Also the number of layers is much larger than the allowed minimum, so that the assumption of a nearly constant computation time for the near formula is justified.

To improve the scaling, one can increase the number of processors, which enters as $1/n_{proc}$ in all the relevant contributions to Eqn. 3.33. Therefore the computation time scales ideally as

$$T_{ideal}^{n_{proc}}(N) = \mathcal{O}(N^{5/3}/n_{proc}) . \tag{3.36}$$

Recalling the discussion in the parallelisation section, a typical implementation will not allow the number of processors to be larger than the number of particles. From this one obtains a lower bound for the computation time scaling of a parallel implementation of MMM2D, which is given by

$$T_{ideal}^{max}(N) = \mathcal{O}(N^{4/3}) . \tag{3.37}$$

The Ewald type methods for two dimensional periodicity consist of plain particle pair force evaluations. A parallelisation of these methods requires a full communication of the particle data from all nodes to all nodes, which makes a parallelisation ineffective at all.

However, for the $3d$–Ewald method one can make a similar estimation. If the link cell algorithm is used for the short ranged part, the number of processors will be smaller than the number of cells. The number of cells scales like $\sqrt{N}$, therefore the maximal possible number of processors that can be used effectively is $\sqrt{N}$, leading to a linear computation time. This shows again that the case of three dimensional periodicity allows for more efficient algorithms than the case of two dimensional periodicity.

Two dimensionally periodic boundary conditions are used primarily for studying surface effects. Therefore the box form plays an important role and the shape factor θ might be quite different from 1. One expects a strong dependency of the computation time on θ, as e. g. for $\theta = \infty$, i. e. a planar system, the scaling of the method increases to $\mathcal{O}(N^2)$. For a constant number of particles the second term of Eqn. (3.33) is dominant for large B. But since all other terms have a prefactor of N at least, and $N >> B$ for reasonable parameters (since otherwise the first two terms scale like

worse than $\mathcal{O}(N^3)$), this term will be small compared to the others. Therefore the important contributions to the computation time are

$$\mathcal{O}\left(\theta B^2\right) + \mathcal{O}(1/B)\,, \tag{3.38}$$

which is minimised by

$$B^N_{ideal}(\theta) = \mathcal{O}(\theta^{-1/3})\,. \tag{3.39}$$

The resulting computation time for a constant number of particles is

$$T^N_{ideal}(\theta) = \mathcal{O}(\theta^{1/3})\,. \tag{3.40}$$

Note that this calculation leaves out the fact that for very large θ the far formula will no longer be used, so that the computation time does not diverge with increasing θ as the scaling suggests. On the other hand for small θ B might reach the lower limit given by Eqn. (3.26), so that the scale is only valid for intermediate values of θ.

For a real simulation the simulation box size is normally not fixed, but instead one keeps the density of the system constant. In two dimensional periodic systems, this means that λ_x and λ_y both scale like $\mathcal{O}(\sqrt{N})$, since the height of the system is determined by physical constraints, e. g. the film thickness of a thin film. Unfortunately this leads to a scaling of

$$\mathcal{O}(B^2N^2) + \mathcal{O}(B^3) + \mathcal{O}(N^2/B) + \mathcal{O}(N)\,, \tag{3.41}$$

which is optimised by a constant layer number B leading to a unfavourable scaling of $\mathcal{O}(N^2)$. Even worse, using multiple processors will not improve the scaling as the number of nodes is bounded by the constant B. For increasing all three spatial dimensions simultaneously, θ is invariant and therefore the scaling stays the same.

The Ewald type methods for two dimensionally periodic systems do not show a similar effect, since the computation time does not depend critically on the shape of the box. Especially the computation time scaling for constant box size or constant density is identical, namely $\mathcal{O}(N^2)$.

Although we know the optimal proportionality of B, its prefactor has to be tuned to the underlying hardware. One can choose the optimal B by measuring the computation time spent for a fixed number of time steps during the simulation. From time to time one increments or decrements B to see whether a better performance can be achieved. The error formulas assure that this will not have any negative impact on the precision of the method.

In the considerations above it was also not taken into account that the height of the layers has to be larger than the range of the other short ranged interactions, especially the diameter of the involved particles. This poses an upper bound on the number of layers. In a typical simulation of a thin film or membrane the system will be between 20 and 100 particle diameters high, so that this limit is not critical.

3.3.5 Numerical Demonstration

In this section we give some numerical results from the implementation of the MMM2D algorithm used in the ESPResSo simulation package. The code implements the parallelised algorithm as described. The cell structure and the link–cell structure coincide,

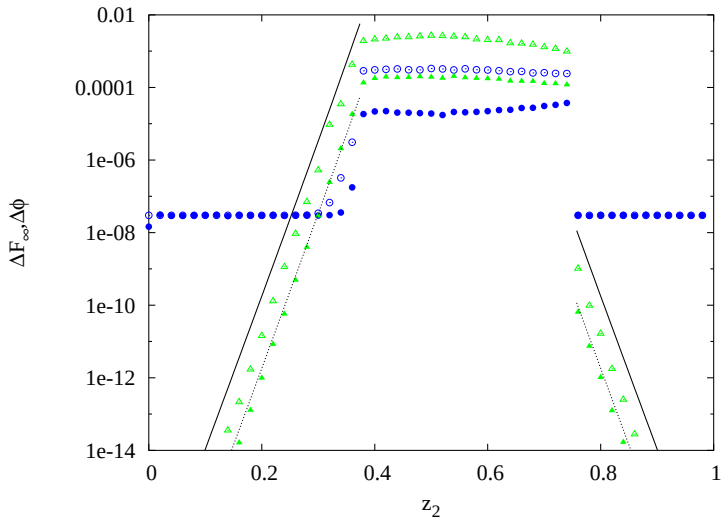


Figure 3.1: Absolute force component error $\Delta F_\infty = \max(|\Delta F_x|, |\Delta F_y|, |\Delta F_z|)$ (triangles) and energy error $\Delta\phi$ (circles) as a function of the z–position of the randomly placed particle. For both the force and energy error the open symbols give the maximal error that occurred, the filled symbols the average error. The solid line shows the force error estimate given by Eqn. (3.19), the dashed line the energy error estimate given by Eqn. (3.18).

i. e. the code cannot handle interaction ranges larger than $1/4 \max(\lambda_x, \lambda_y)$. All single node computations presented in the following were performed on a single AMD Athlon64 3000+ processor with 512 MB DDR400 RAM. The parallel computations were performed on an IBM Regatta pServer using up to 16 Power4 1.4 GHz processors.

2–particle systems

The first presented here is the non–uniform error distribution of MMM2D that was stated in Sec. 3.3.2. For Figure 3.1 we used a cubic simulation box of unit length containing two particles, one located at $(0, 0, 0.5)$, the other randomly placed in the box. The electrostatic interaction was calculated 100000 times, using MMM2D tuned to an maximal force component error of 10^{-2} with $B = 8$ layers, and using MMM2D with an error 10^{-17} and $B = 4$ layers. The results of the second calculation were used as "exact" results to compare with. Due to the different numbers of layers for $z \in [0.25, 0.375)$ or $z \in [0.75, 1)$ for the exact calculation the near formula is used and for the other the far formula. This demonstrates that the far formula and the near formula indeed coincide up to numerical errors.

Fig. 3.1 shows clearly that the error distribution is highly non–uniform. While the near formula due to its adaptive implementation of the Bernoulli and polygamma sums as described in Sec. (3.3.2) has a nearly uniform error distribution with a maximal error close to the limit, the far formula reaches its maximal error only at the point of closest approach 0.375. From this point down the error drops exponentially as predicted. The steepness of the exponential drop is given by R, which is the reason

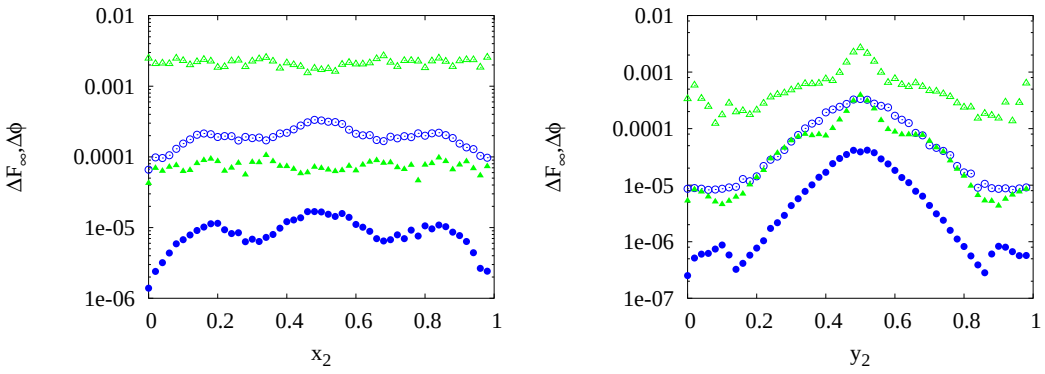


Figure 3.2: Same as Fig. 3.1, but here the error distribution is given as a function of the x–coordinate (left) and y–coordinate (right).

why a precision of only 10^{-2} was chosen, since otherwise the exponential drop would not have been visible anymore. At the right side, at 0.75, the situation is even worse, since the fixed particle is placed as far as possible from the upper border of its layer. If the fixed particle were placed close to 0.625, the situation would have been exchanged, i. e. the error for small z would be excessively small. In principle it would be possible to take less (p, q)–vectors into account for higher z–distances, but this would actually *increase* the computation time as the calculation of the $\Xi^{(l/h,s/c,s/c)}$ gets more complex. The $\Xi^{(\pm,s/c,s/c)}$ would not only have to be summed up, but also some from lower layers would have to be subtracted again. To this aim additional communications between non–adjacent processors would be necessary, which is clearly unfavourable.

The error in the potential is lower than the error of the forces by an order of magnitude as predicted. The plateau reached at a precision of $5 \cdot 10^{-8}$ reflects the precision of the self–energy. One should be aware that the error distribution does not average out since for particles near the z–borders of the simulation box more interactions will be treated by the far formula with excessive precision than for a particle in the centre, so that the overall precision is better for the particles at the z–borders. As long as the errors are tuned to be sufficiently low (e. g. an order of magnitude below the thermal energy), this will not do any harm. The error distribution along the y–axis is also non–uniform due to the asymmetric construction of the near formula. But since this error is translationally invariant in a periodic coordinate, it averages out, and will not bias the results.

In the time scale considerations above, the error estimate for the far formula has a central place. Fig. (3.1) shows that the estimates τ_F (for the forces) and τ_E (for the energy) are accurate. They both slightly overestimate the maximal errors as measured, but are close enough to the numerical results. Therefore the timing considerations should not be affected.

Many particle systems

Having laid the theoretical foundations of the time scaling in Sec. 3.3.3, it will be shown now that the theoretical scaling can be achieved in a real computation. The timings presented in the following always give the time for a single force evaluation.

The computational structure of the potential evaluations is identical, although the final summation for the far formula uses three times less operations. In general the time for a potential evaluation is around 75% of the computation time of a force evaluation.

Three different scalings with the number of particles were presented in the previous section: For constant box length a computation time scaling of $\mathcal{O}(N^{5/3})$ is expected, while for constant density the scaling is $\mathcal{O}(N^2)$, and finally the minimal $\mathcal{O}(N^{4/3})$ scaling on a multiprocessor system. To check these, three corresponding force computations were performed using MMM2D tuned to a maximal pairwise force error of 10^{-4}.

Run 1: N particles of unit charge, half of them with positive sign and half of them with negative sign, are placed randomly in a simulation box of constant size $10 \times 10 \times 10$. The simulation was performed on a single Athlon64 processor.

Run 2: The N particles are placed randomly in a simulation box of constant box height $h = 10$ and constant density $\rho = 0.5$. The simulation was performed again on the single Athlon64 processor.

Run 3: The same as run 1, but instead of the single AMD processor, $1, 2, 3, \ldots 16$ processors of the IBM pServer were used.

Fig. (3.3) shows the computation time for different numbers of layers and particles from run 1. One can see that the computation time shows a clean shallow minimum, which can be easily found during a production run by e. g. increasing the number of layers until the performance drops. Note that for 1000 particles the minimal computation time is obtain for 15 layers, but the computation time for 14 and 16 layers differs from the minimal computation time by less than the accuracy of the time measurements. Therefore numerical simulations can determine the optimal number of layers only up to a certain range. For practical applications this is very convenient since one does not necessarily have to determine the optimal number of layers, and still obtains a nearly optimal computation time.

Figs. (3.4) and (3.5) show the computationally optimal number of layers and the achievable computation time for different numbers of particles for runs 1 and 2. The

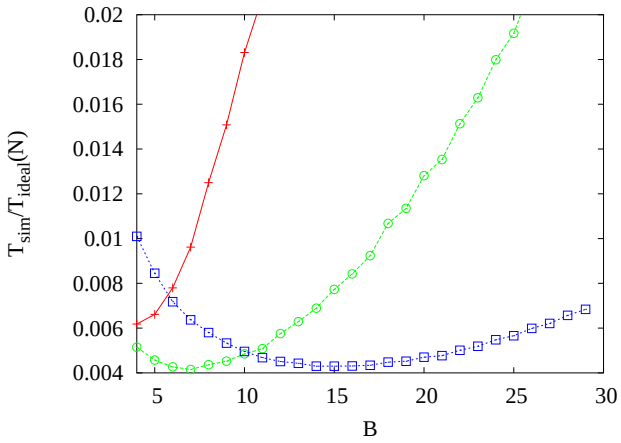


Figure 3.3: Computation times of a MMM2D evaluation for different numbers of layers B. The number of particles were 10 (red plus symbols), 100 (green circles) and 1000 (blue rectangles). The computation times are given in multiples of $T_{ideal}(N) = N^{\frac{5}{3}}$ms.

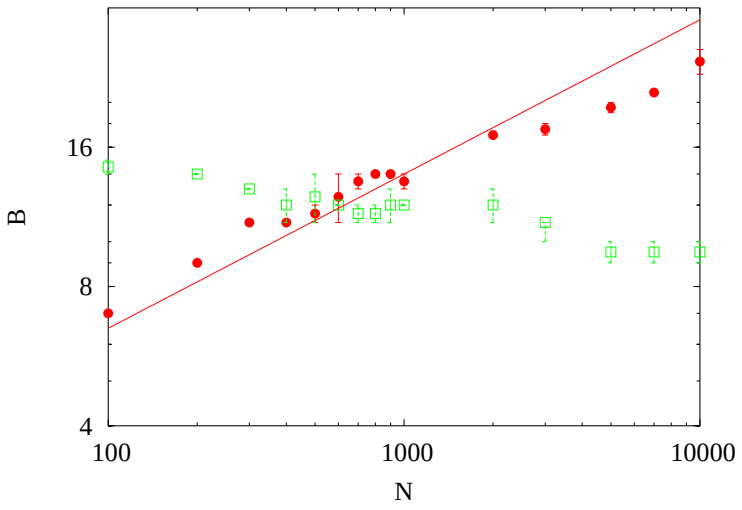


Figure 3.4: Optimal number of layers B of MMM2D for different numbers of particles N. The filled red circles denote results from run 1, the open blue ones results from run 2. The solid red line marks a fit of $B_{ideal}(N) = \mathcal{O}(N^{1/3})$ (see Eqn. (3.34)) to the corresponding data. The errorbars denote the range of numbers of layers for which the computation time is at most 10ms larger than the optimum, which is the resolution of the time measurement.

computation time predictions of the previous sections, and to a lesser extent also the predictions for the optimal number of layers, fit well to the numerical results.

However, above 3000 particles, for both runs the number of layers shifts to a lower value, while the computation time shifts to slightly higher prefactor. This can be understood from the fact that the L2 cache of the Athlon64 is 512k large, which is equivalent to 3000 particles in ESPResSo. For more than 3000 particles, more and more parts of the particle data have to be loaded from the slower main memory instead of the processor cache, which hits primarily the calculation of the far formula, which has to access the particle data $\mathcal{O}(R^2)$–times. The algorithm responds to this by decreasing the amount of computation time spent in the far formula.

For small numbers of particles in run 2 the optimal number of layers B increases. This can be understood from Eqn. (3.26), which gives the lower limit $4u_y h = 4h/\sqrt{N/\rho/h}$ for the near formula to be valid. For 1000 particles this gives $B > 3$, which is far from the optimal number of layers $B = 12$, but for 100 particles, the limit is $B > 9$. Close to this limit the computation time of the near formula starts to increase, which the algorithm compensates by increasing the number of layers.

The results from run 3 are shown in Figs 3.6 and 3.7. Note that the single processor computation times of the Power4 processor are roughly 80% higher than the computation times using the AMD processor for large particle numbers. For small numbers of particles, which fit into the L2 cache of the Athlon64, the computation time is even 125% higher. This difference is even higher than what could be expected from the processor clocks, which are 2GHz for the Athlon64 and 1.3GHz for the Power4. Since similar differences have been obtained also for other simulations

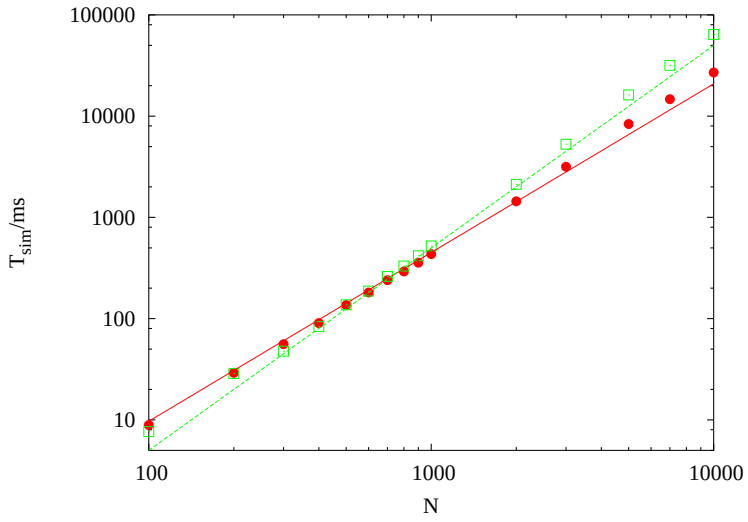


Figure 3.5: Computation times of MMM2D in milliseconds using the optimal numbers of layers as shown in Fig. 3.4. The symbols are the same as in Fig. 3.4, the lines are fits of the predicted scaling of the computation time $T_{ideal}(N) = N^{5/3}$ (solid red, see Eqn. (3.35)) for constant box size and N^2 (dotted blue) for constant density to the corresponding data.

using ESPResSo, e. g. a simple Lennard–Jones fluid [9], this has to be addressed to fundamental architectural differences between the two processor types.

Nevertheless, on the IBM pServer MMM2D still scales as $\mathcal{O}(N^{5/3})$, and one can see that minimal achievable computation time scales as $\mathcal{O}(N^{4/3})$, although the overhead for small numbers of particles is comparatively large. This is simply due to the fact that the minimal computation time is achieved with the minimal possible number of layers, i. e. a single cell per processor. For the calculation of the near formula, the particle data of each processor has to be communicated to its neighbor. In total, nearly the full particle data has to communicated once per energy evaluation, generating a large communication time overhead.

The efficiency of an parallel algorithm is the total computation time of all processors divided by the computation time of one processor, i. e. $T_{cpu} n_{proc} / T_{cpu,n_{proc}=1}$. Theoretically, the computation time scales like $1/n_{proc}$ (see Eqn. 3.36), so that the efficiency should be 1. The efficiency obtained from run 3 is nearly as good, roughly 90% with 16 processors. For 10000 particles on two processors the efficiency is roughly 100%.

The predicted logarithmic dependency of the computation time from the error bound is shown in Fig. 3.8. Here 500 particles were placed randomly in a $10 \times 10 \times 10$ simulation box and the force calculated using MMM2D at varying error bounds. The number of layers was chosen separately for each error bound such that the computation was minimal. Since the computation of both the near formula and the far formula depend logarithmically on the error bound, one expects the number of layers to be constant with respect the error bound. Indeed the number of layers was 10 except for small error bounds below 1e-6, where the optimal number of layers gradually increases

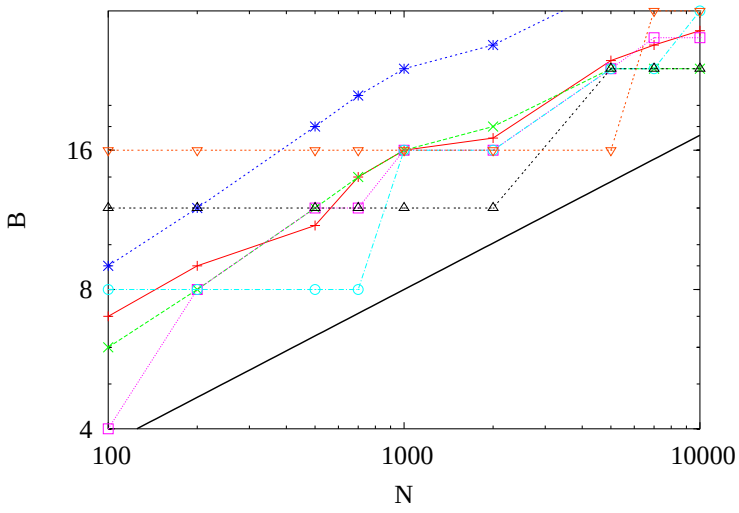


Figure 3.6: Optimal number of layers B of MMM2D for different numbers of particles N and numbers of processors. The numbers of processors were 1 (red plusses), 2 (green stars), 3 (blue stars), 4 (magenta squares), 8 (cyan circles), 12 (black triangles up) and 16 (orange triangle downs). The implementation of MMM2D in ESPResSo requires an equal number of cells per processor, so that for 16 processors the number of layers is necessarily a multiple of 16. The black line marks the predicted optimal scaling of the number of layers $B_{ideal}(N) = \mathcal{O}(N^{1/3})$ (see Eqn. (3.34)).

to 14 for 10^{-2}. The reason for this is that the far formula with its single cutoff R can be tuned much more precisely than the near formula. Therefore with decreased error bound the computation time decreases more for the far formula than for the near formula, which leads to a higher optimal number of layers.

In Fig. 3.9 we demonstrate the scaling of the optimal computation time with increasing box length at constant particle number. Fig. 3.9 shows the computation time and number of layers of a simulation with 1000 randomly placed particles in a box with fixed $\lambda_x = \lambda_y = 10$ and various h. For large h and therefore B the computation time does no longer scale as predicted, since here the $\mathcal{O}(B^3)$ part of the layer communication becomes visible. The obvious wave structure on top of the scaling in the optimal layer number cannot be explained easily.

The predicted computation time scalings can be reproduced well by the implementation of MMM2D in ESPResSo. The optimal B only loosely follows the predicted scalings, so that its value cannot be determined a priori very well. The suggested trial and error method seems to be the only viable method. This is supported by the fact that the computation time minimum in B is very shallow and the existence of rigorous error bounds, which asserts that changing the number of layers will not influence the outcome of the simulation.

In [4, 6] it was shown that MMM2D is faster than all other methods for two dimensionally periodic systems known so far, and is the first one featuring a rigorous error bound. Together with the parallelisation this method is the method of choice for high precision requirements (e. g. below 10^{-8}) or moderate numbers of particles.

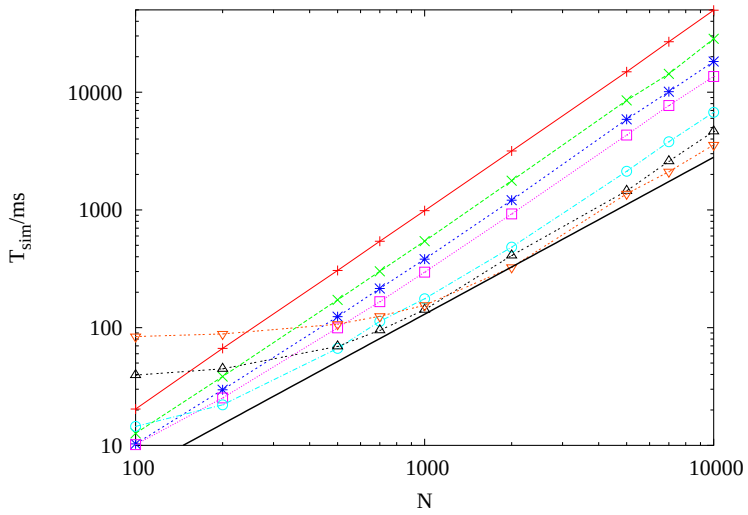


Figure 3.7: Computation times of MMM2D in milliseconds using the optimal number of layers as shown in Fig. 3.6. The symbols are the same as in Fig. 3.6, the black line marks the predicted minimal computation time scaling $\mathcal{O}(N^{4/3})$.

On a single processor for around 500 particles the computation time grows above $0.1s$. For a real simulation, the computation time should not be much larger, since a million time steps will take around one day. For low precision requirements and large numbers of particles, another method, ELC, will be presented in the following.

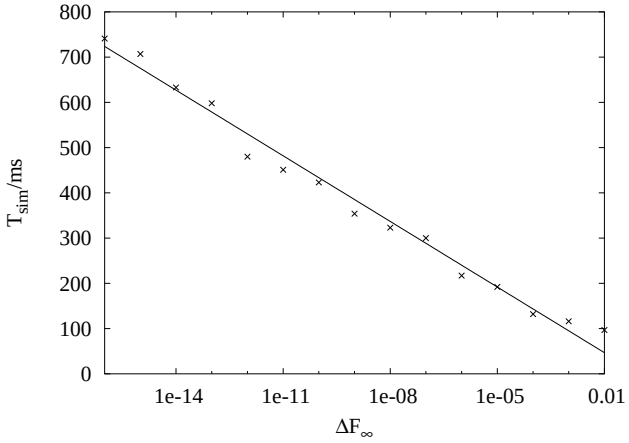


Figure 3.8: Computation times of a MMM2D force evaluation with different error bounds ΔF_∞ for the force. The solid line is a fit of $-\log(\epsilon)$ to the data.

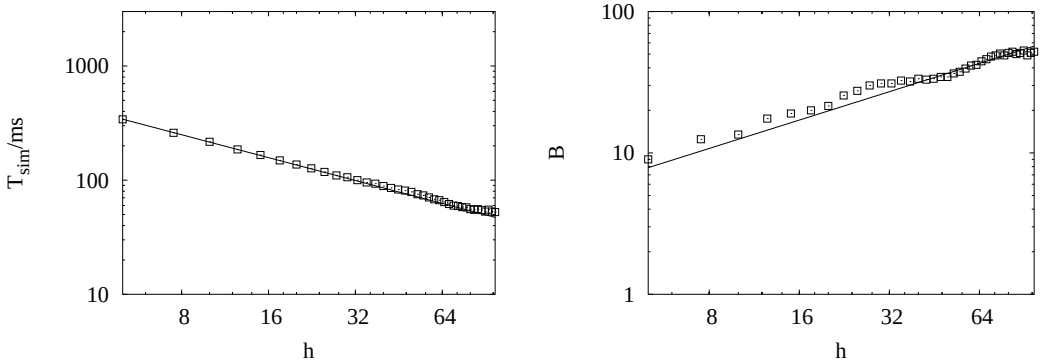


Figure 3.9: Optimal setting of the number of layers B (right graph) and the corresponding computation times (left graph) for different box lengths h for 1000 particles on a single processor. The solid lines are fits of the predicted relations $B^N_{ideal}(\theta) = \theta^{-1/3} \sim h^{2/3}$ (right graph, see Eqn. (3.39)) and $T^N_{ideal}(\theta) = h^{-2/3}$ (left graph, see Eqn. (3.40)).

4 ELC — fast electrostatics for two dimensional periodicity

For a large number of particles the computation time scaling should not have a power of N larger than 1. So far such a scaling is only known for methods for three dimensional periodicity. Therefore early attempts tried to use such a method for the case of two dimensional periodicity. Since the system is overall charge neutral, it is obvious to replicate the system also in the non–periodic axis and leave a sufficiently large gap. This approach was seen early to produce significant errors [102]. Yeh and Berkowitz [102] suggested to change the summation order to a summation layer by layer, which indeed gives a reliable method.

In the following a new expression, called the electrostatic layer correction (ELC) term, for the error produced by the artificial images will be derived. From this expression one can estimate the error produced by the method proposed by Yeh et al., and one can tune the gap size according to the desired accuracy. But that is only a byproduct. Its primary advantage is that it can be used to subtract the contributions of the unwanted image layers, which can be evaluated with a computation time proportional to N. Therefore it can be combined with any conventional method for three dimensions without decreasing the computation time scaling, allowing one to use a much smaller gap between the layers, thereby speeding up the computation time considerably. The ELC method has been published previously in Refs. [5, 16].

4.1 Changing the summation order

We consider the same system of two dimensional periodicity contained in a box of dimension $\lambda_x \times \lambda_y \times h$. So far the particles are only replicated periodically in x and y. The basic idea is now to expand this slab system in the non-periodic z–coordinate to a system with periodicity in all three dimensions, with a period of $\lambda_z > h$, which leaves an empty gap of height $\delta = \lambda_z - h$ above the particles in the simulation box. Fig. 4.1 shows a schematic view of the resulting system. δ will be called gap size in the following.

Since the electrostatic potential is only finite if the total system is charge neutral,

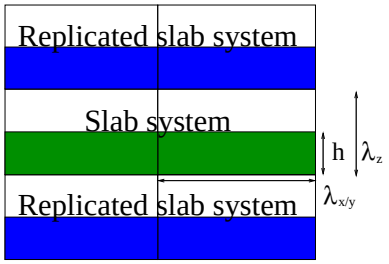
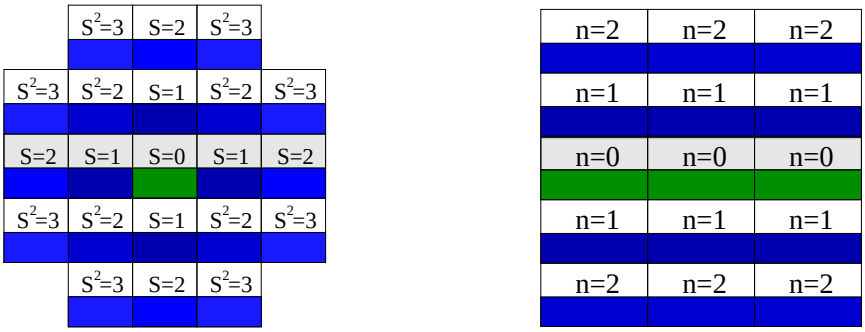


Figure 4.1: Schematic representation of a fully periodically replicated slab system



(a) Spherical summation order. S is the length of the box offset.

(b) Slab–wise summation order. n is the z offset of the box, the spherical summation order in the x, y–plane is not shown.

Figure 4.2: Schematic view of two different summation orders.

the additional image layers (those layers above or below the original slab system) are charge neutral, too. Now let us consider the n^{th} image layer which has an offset of $n\lambda_z$ to the original layer. If $n\lambda_z$ is large enough, each particle of charge q_j at position $(x_j, y_j, z_j + n\lambda_z)$ and its replicas in the x, y-plane can be viewed as constituting a homogeneous charged sheet of charge density $\sigma_j = \frac{q_j}{\lambda_x \lambda_y}$. The potential of such a charged sheet at distance z is $2\pi\sigma_j|z|$. Now we consider the contribution from a pair of image layers located at $\pm n\lambda_z$, $n > 0$ to the energy of a charge q_i at position (x_i, y_i, z_i) in the central layer. Since $|z_j - z_i| < n\lambda_z$, we have $|z_j - z_i + n\lambda_z| = n\lambda_z + z_j - z_i$ and $|z_j - z_i - n\lambda_z| = n\lambda_z - z_j + z_i$, and hence the interaction energy from those two image layers with the charge q_i vanishes by charge neutrality:

$$2\pi q_i \sum_{j=1}^{N} \sigma_j (|z_j - z_i + n\lambda_z| + |z_j - z_i - n\lambda_z|) = 4\pi q_i n\lambda_z \sum_{j=1}^{N} \sigma_j = 0\,. \tag{4.1}$$

The only errors occurring are those coming from the approximation of assuming homogeneously charged, infinite sheets instead of discrete charges. This assumption should become better when increasing the distance $n\lambda_z$ from the central layer.

However, in a naive implementation, even large gap sizes will result in large errors [102]. This is due to the order of summation for the standard Ewald sum, which is spherical. This order implies that with increasing shell cutoff S the number of image shells grows faster than the number of shells of the primary layer, namely $\mathcal{O}(S^3)$ versus $\mathcal{O}(S^2)$ (see Fig. 4.2(a)). In other words, we include the unwanted terms faster than the actually wanted terms. Also the image layers are not really infinite charged sheets but are truncated due to the cut-off. Yeh and Berkowitz [102] already suggested that this problem can be solved by changing the order of summation. Smith has shown that by adding to the Ewald energy the term

$$E_c = 2\pi M_z^2 - \frac{2\pi M^2}{3}\,, \tag{4.2}$$

where M is the total dipole moment as given by Eqn. (2.10), one obtains the result of a slab–wise summation instead of the spherical limit [87]. Slab–wise summation

refers to the sum $\sum_{|n|\geq 0} E_l(n)$, where $E_l(n)$ denotes the energy, calculated in spherical summation order (in the generalised sense), resulting from the image layer with shift $n\lambda_z$ in the z–coordinate. Technically this is the order where one treats the original layer first and then add the image layers grouped in symmetrical pairs (see Fig. 4.2(b)). Obviously this summation order fits much better to the charged sheet argument given above. Although this is a major change in the summation order, the difference given by Eqn. (4.2) is a very simple term. In fact, Smith shows that changes of the summation order always result in a difference that depends only on the total dipole moment.

Applying this slab–wise summation order, Yeh and Berkowitz stated that a gap size of at least h is normally sufficient to obtain an moderately accurate result. Therefore the result of a standard Ewald method plus the shape–dependent term given by Eqn. (4.2), which will be called the *slab–wise method*, can be used to obtain a good approximation to the result for the slab geometry with the same computational effort as for the underlying $3d$–Ewald method (no matter if a simple or sophisticated method is used). One drawback is that no theoretical estimates exist for the error introduced by the image layers. Therefore one might be forced to use even larger gaps to assure that no artifacts are produced by the image layers. One simple deducible artifact is that the pairwise error will be position dependant. Particles in the middle of the slab will see no effect of the image layers due to symmetry, and particles near the surface will encounter for the same reason the largest errors, which is definitely an unwanted feature for studying surface effects. Therefore averaging error measures like the commonly used RMS force error should not be applied without additional checks for the particles near the surfaces.

The other drawback is that normally the box now will have a significantly larger λ_z/λ_x rsp. λ_z/λ_y. But for Ewald methods the computation time is proportional to these fractions [16]. This is easy to see as the number of k–space vectors in the z direction must be proportional to λ_z to maintain a fixed resolution and therefore error. It is verified experimentally that a gap of at least h is needed for a nearly cubic simulation box. For a cubic system $h = \lambda_x = \lambda_y$ therefore the computation time will be increased by a factor of around two.

Nevertheless because of the bad scaling of other methods for slab geometries which is at best $\mathcal{O}(N^{5/3})$ as for MMM2D, for particle numbers above $N \approx 1000$ using slab–wise methods is a great improvement.

4.2 The electrostatic layer correction term

We will now derive a term that allows to calculate the *exact* contribution of the image layers very efficiently, which will be called the electrostatic layer correction (ELC) in the following. The layer correction itself will be using the far formula of MMM2D. For the following analysis there is no special restriction on h except for $h < \lambda_z$, which is always true for a finite number of particles.

We start with the formal definition of the Coulomb energy of the slab system

$$E = \frac{1}{2} \sum_{S=0}^{\infty} \sum_{k^2+l^2=S} \sum_{i,j=1}^{N}{}' \frac{q_i q_j}{|p_i - p_j + n_{kl}|} \,. \tag{4.3}$$

We now expand the system to a fully three-dimensional periodic system, where λ_z determines the period in the z-coordinate as for a truly three dimensional periodic system. We can rewrite the energy as

$$E = E_s + E_c + E_{lc}\,, \tag{4.4}$$

where

$$E_s = \frac{1}{2}\sum_{S=0}^{\infty}\sum_{k^2+l^2+m^2=S}\sum_{i,j=1}^{N}{}' \frac{q_i q_j}{|p_i - p_j + n_{klm}|}\,. \tag{4.5}$$

denotes the standard three-dimensional Coulomb–sum with spherical limit. To evaluate this expression one can use any of the efficient algorithms, starting with the classical Ewald summation up to modern methods like fast multipole methods [57] or mesh based algorithms [19]. E_c again denotes the shape–dependent term given by Eqn. (4.2) and finally

$$E_{lc} = -\frac{1}{2}\sum_{m>0}\sum_{m=\pm m}\sum_{S=0}^{\infty}\sum_{k^2+l^2=S}\sum_{i,j=1}^{N}\frac{q_i q_j}{|p_i - p_j + n_{klm}|}\,. \tag{4.6}$$

denotes the contribution of the image layers, for which we are going to derive a new expression in the following.

We start with the expression for the energy induced by an image layer at z–offset $m \neq 0$:

$$E_l(m) = -\frac{1}{2}\sum_{S=0}^{\infty}\sum_{k^2+l^2=S}\sum_{i,j=1}^{N}\frac{q_i q_j}{|p_i - p_j + n_{klm}|}\,. \tag{4.7}$$

Since $m \neq 0$ this expression can be easily evaluated using the far formula, and we obtain

$$E_{lc}(m) = -\frac{1}{2}\sum_{i,j=1}^{N} q_i q_j \phi(p_i - p_j + n_{00m})\,, \tag{4.8}$$

where ϕ is given by

$$\begin{aligned}\phi(x,y,z) = {} & 4u_x u_y \sum_{p,q>0}\frac{e^{-2\pi f_{pq}|z|}}{f_{pq}}\cos(\omega_p x)\cos(\omega_q y) + \\ & 2u_x u_y \sum_{p>0}\frac{e^{-2\pi f_p|z|}}{f_p}\cos(\omega_p x) + 2u_x u_y \sum_{q>0}\frac{e^{-2\pi f_q|z|}}{f_q}\cos(\omega_q y) - 2\pi u_x u_y |z|\,.\end{aligned} \tag{4.9}$$

For now we only have a formula for the contribution of one image layer, so we still have to sum over all m. This task can be performed analytically. The terms $2\pi u_x u_y|z|$ can be omitted since they are exactly the homogeneous sheet potential and we have seen before that this cancels out for charge neutral systems (see Eqn. (4.1)). The summation over m of the remaining sums over (p,q) is performed using the geometric series (as these sums are absolutely convergent, exchanging the summation over m and the summations over (p,q) is possible). For $m > 0$ we obtain, ignoring the terms independent of m,

$$\sum_{m\geq 1} e^{-2\pi f_{pq}|z+m\lambda_z|} = \frac{e^{-2\pi z}}{e^{2\pi f_{pq}\lambda_z} - 1}\,, \tag{4.10}$$

for $m < 0$

$$\sum_{m\leq -1} e^{-2\pi f_{pq}|z+m\lambda_z|} = \frac{e^{2\pi z}}{e^{2\pi f_{pq}\lambda_z}-1} \tag{4.11}$$

since $|z| \leq \lambda_z$.

Combining the terms for $\pm m$ again we obtain

$$E_{lc} = \sum_{i,j=1}^{N} q_i q_j \psi(p_i - p_j)\,, \tag{4.12}$$

where

$$\begin{aligned}\psi(x,y,z) =& 4u_x u_y \sum_{p,q>0} \frac{\cosh(2\pi f_{pq} z)}{f_{pq}(e^{2\pi f_{pq}\lambda_z}-1)} \cos(\omega_p x)\cos(\omega_q y) + \\ & 2u_x u_y \sum_{p>0} \frac{\cosh(2\pi f_p z)}{f_p(e^{2\pi f_p \lambda_z}-1)} \cos(\omega_p x) + \\ & 2u_x u_y \sum_{q>0} \frac{\cosh(2\pi f_q z)}{f_q(e^{2\pi f_q \lambda_z}-1)} \cos(\omega_q y)\,.\end{aligned} \tag{4.13}$$

The forces can be obtained from that by simple differentiation since the sums are absolutely convergent. Although the form in Eqn.(4.13) has a much better convergence than the original form in Eqn.(4.6), its main advantage is a linear computation time with respect to the number of particles N. This is achieved similar to the far formula of MMM2D, using the addition theorems for the cosine rsp. sine hyperbolicus.

4.3 Implementation

The implementation is very similar to MMM2D. This time the per–particle–terms are

$$\chi_i^{(c/s,c/s,c/s)} = q_i \cosh / \sinh(2\pi f_{pq} z_i) \cos / \sin(\omega_p x_i) \cos / \sin(\omega_q y_i) \tag{4.14}$$

for the (p,q) vectors with $p, q \neq 0$ and

$$\begin{aligned}\chi_i^{(x,c/s,c/s)} &= q_i \cosh / \sinh(2\pi f_p z_i) \cos / \sin(\omega_p x_i)\,, \\ \chi_i^{(y,c/s,c/s)} &= q_i \cosh / \sinh(2\pi f_q z_i) \cos / \sin(\omega_q y_i)\end{aligned} \tag{4.15}$$

for the $(p,0)$ and $(0,q)$ vectors. The indices in the obvious way determine which of the functions cosine (hyperbolicus) or sinus (hyperbolicus) are used. For the energy

one has to evaluate

$$
\begin{aligned}
E_{lc} = 4u_x u_y \sum_{p,q>0} \sum_{i,j=1}^{N} \frac{1}{(e^{2\pi f_{pq} L_z} - 1) f_{pq}} \Big(& \\
& \chi_i^{(ccc)} \chi_j^{(ccc)} + \chi_i^{(csc)} \chi_j^{(csc)} + \chi_i^{(ccs)} \chi_j^{(ccs)} + \chi_i^{(css)} \chi_j^{(css)} - \\
& \chi_i^{(scc)} \chi_j^{(scc)} - \chi_i^{(ssc)} \chi_j^{(ssc)} - \chi_i^{(scs)} \chi_j^{(scs)} - \chi_i^{(sss)} \chi_j^{(sss)} \Big) + \\
2u_x \sum_{p>0} \sum_{i,j=1}^{N} \frac{1}{(e^{2\pi f_p L_z} - 1) f_p} \Big(& \\
& \chi_i^{(xcc)} \chi_j^{(xcc)} + \chi_i^{(xcs)} \chi_j^{(xcs)} - \chi_i^{(xsc)} \chi_j^{(xsc)} - \chi_i^{(xss)} \chi_j^{(xss)} \Big) + \\
2u_y \sum_{q>0} \sum_{i,j=1}^{N} \frac{1}{(e^{2\pi f_q L_z} - 1) f_q} \Big(& \\
& \chi_i^{(ycc)} \chi_j^{(ycc)} + \chi_i^{(ycs)} \chi_j^{(ycs)} - \chi_i^{(ysc)} \chi_j^{(ysc)} - \chi_i^{(yss)} \chi_j^{(yss)} \Big) .
\end{aligned}
\tag{4.16}
$$

Similar to MMM2D the summations over i rsp. j can be calculated beforehand. To this aim every node calculates the sum over the local particles, then this value is added up over all nodes and then the sum distributed back. The latter is necessary for the force calculation, where no summation over i occurs, for the potential it is sufficient to only gather the data on the master node and do the final summation over (p, q) only there. The summation and distribution of the $\chi_i^{(c/s,c/s,c/s)}$ is implemented efficiently in MPI, where it is termed a *reduce all* operation. Together with a fast method for three dimensional periodicity such as P3M or MMM we have a parallel algorithm for two dimensional periodicity with a scaling of $\mathcal{O}(N \log N)$.

In pseudo code the calculation for one (p, q) vector of the ELC term on a single node looks like this:

$\chi^{(s/c,s/c,s/c)} = 0$
for each particle j
calculate $\chi_j^{(s/c,s/c,s/c)}$
$\chi^{(\pm,s/c,s/c)} += \chi_j^{(\pm,s/c,s/c)}$
end
for each particle j
calculate particle interaction from $\chi_j^{(s/c,s/c,s/c)} \chi^{(s/c,s/c,s/c)}$
end

In a multiprocessor environment the *reduce all* operation would be placed between the two loops.

4.4 Error estimates

The next step to do is the adaption of the error formulas for the far formula to our newly developed ELC formula given by Eqn. (4.13). We will show that using ELC the errors will be highest for the particles near the borders of the simulation box instead of lowest as for MMM2D. Therefore again the maximal pairwise error seems a more reasonable error estimate than the conventionally used RMS force error. But since for the widely used Ewald methods the RMS force error is the standard error measure, this error measure plays a more prominent role for ELC. It was already described for MMM2D how such an error measure can be estimated from the maximal pairwise error.

While the error bounds for MMM2D were only used to tune the algorithm, the error estimates for ELC can also be used to obtain an error bound for the slab–wise method from Ref. [102], and hence one can determine "a priori" the necessary gap size to reach a preset precision. Therefore we also have to deal with small cutoffs, especially the case when no terms of E_{lc} are added.

The summation is performed only over all (p,q) vectors contained in Γ_R as defined in Eqn. (3.17). An upper bound for the absolute value of the summands is

$$\left|\frac{\cosh(2\pi f_{pq}z)}{f_{pq}(e^{2\pi f_{pq}\lambda_z}-1)}\cos(\omega_p x)\cos(\omega_q y)\right| \leq e^{-2\pi f_{pq}\lambda_z}\frac{\cosh(2\pi f_{pq}z)}{f_{pq}(1-e^{-2\pi f_{pq}\lambda_z})} \leq e^{-2\pi f_{pq}\lambda_z}\frac{\cosh(2\pi f_{pq}h)}{f_{pq}(1-e^{-2\pi f_{pq}\lambda_z})}\,. \quad (4.17)$$

The sum over all these upperbounds can then be performed similar to the error estimate for MMM2D. Of course because the cosine hyperbolicus is monotonous, one could use any larger value for h. This is for example necessary in a priori estimations. Using this we find the upper bound for the maximal pairwise error similar to Eqn. (3.18) as

$$\tau_E := \frac{1/2+\frac{u_x+u_y}{\pi R}}{e^{2\pi R\lambda_z}-1}\left(\frac{\exp(2\pi Rh)}{\lambda_z-h}+\frac{\exp(-2\pi Rh)}{\lambda_z+h}\right), \quad (4.18)$$

and the error estimate for the forces is given by

$$\tau_F := \frac{1}{2(e^{2\pi R\lambda_z}-1)}\left(\left(2\pi R+2(u_x+u_y)+\frac{1}{\lambda_z-h}\right)\frac{\exp(2\pi Rh)}{(\lambda_z-h)}+\left(2\pi R+2(u_x+u_y)+\frac{1}{\lambda_z+h}\right)\frac{\exp(-2\pi Rh)}{(\lambda_z+h)}\right), \quad (4.19)$$

which again is also a weaker bound for the potential.

The error produced by the method of Yeh and Berkowitz can be obtained by using a small R, and adding the upper bound for the (p,q) not yet covered by Γ_R separately. $R=0$ cannot be used since the error estimates are both obviously singular for $R\to 0$. This is due to the fact that f_{pq} is replaced by its lower limit R during the estimation. But $R\sim\max(u_x,u_y)$ is sufficient for the error estimate to be of practical use.

Note that Eqn.(4.17) shows that the error in the potential or the force for a single particle will be largest if it is located near the gap, since there $|z_{ij}|$ will be maximal.

This effect will increase with increasing R. Therefore when using the layer correction one must apply non–averaging error estimates such as the maximal pairwise error. Averaging error estimates such as the RMS force error might be misleading about the error of the particles near the gap, which often are of special interest in simulations of surfaces or thin films.

All our error estimates show that the error drops exponential both with R and λ_z. The decay in R means that it is easy to achieve high accuracies with our layer correction formula, while the decay in λ_z shows that slab–wise methods can achieve good accuracies without increasing λ_z too much. But the error formulas also show that for $\lambda_z >> h$ not the fraction λ_z/h dominates the error, but rather $R\lambda_z$. From the form of Γ_R one can see that for R scales like u_z rsp. u_y. Therefore the actually important fractions are λ_z/λ_x and λ_z/λ_y and *not* λ_z/h as is commonly believed. Also the extreme case of $h = 0$ immediately shows that this cannot be true, as a conventional method for three dimensional periodicity will not deliver exact results for a purely planar system.

If one assumes an Ewald type of method to be used for the three dimensional periodic system, one can estimate the optimal λ_z as follows. The computation time of the Ewald type method is proportional to $\lambda_x\lambda_y\lambda_z$. R is proportional to $1/(\lambda_z - h)$, therefore the computation time spent with ELC is proportional to $\lambda_x\lambda_y/(\lambda_z - h)^2$. The total computation time is therefore minimised by a constant gap size independent of the box dimensions.

4.5 Numerical demonstration

In this section results from the implementation of the layer correction ELC within ESPResSo are shown. As method for the three dimensionally periodic system always P3M is used, tuned to an RMS force error of less than 10^{-4}. Our implementation of P3M is limited to cubic simulation boxes, therefore the simulation box is always cubic in the following with a box length of λ. This also does not allow to present numerical results for a plain implementation of the Yeh and Berkowitz method, as this requires λ_z to be at least four times larger that λ_x or λ_y. But using the argument above the number of k–vectors of P3M will be four times larger so that also the computation time will be around four times larger than the values presented here for the plain P3M.

The robustness of the error estimates was already shown for MMM2D and will therefore not be addressed here again. What will be shown is the predicted error accumulation at the borders of the simulation box. To this aim a cubic simulation box of side length 10 is filled randomly with 1000 charged particles as before, but the z–coordinates range only from 0 to 9, i. e. a gap of size 1 is left empty in the z–direction. For this system the electrostatic interactions were calculated 50 times and compared to a MMM2D calculation tuned to a maximal pairwise error below 10^{-15}. Fig. 4.3 shows the resulting error distribution along the z–axis for different cutoffs R of ELC. Comparing the $h = 9$ and the $h = 5$ case one can see that the overall maximal error drops for all error bounds. But still the error for the $R = 0$ case is unacceptable high. Since our implementation is limited to $\lambda_x = \lambda_y = \lambda_z$, this is no wonder. More interesting is the fact that the error of the cut off layer correction increases

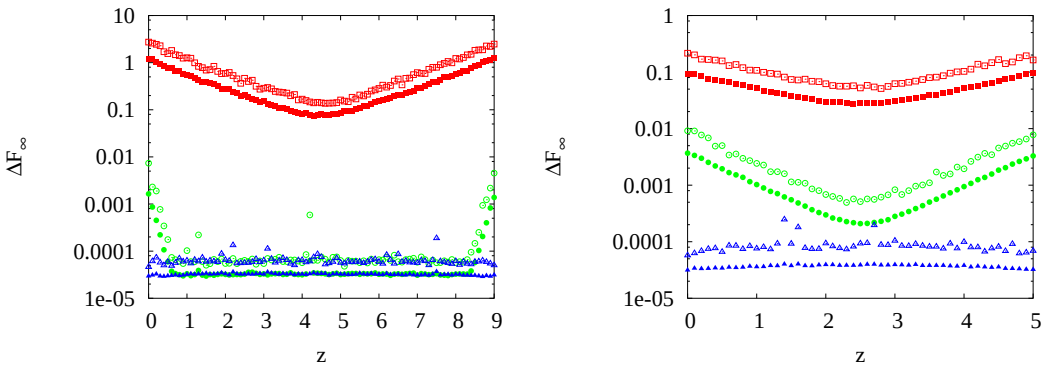


Figure 4.3: Absolute force error ΔF_∞ as a function of the particle z–coordinate for a homogeneous random system of 1000 particles in a box of size $10 \times 10 \times h$ with a system height $h = 9$ (left graph) and 5 (right graph). The red rectangles denote results for $R = 0$, the green circles for a maximal error of 10^{-2}, corresponding to $R = 1$ for $h = 9$ and $R = 0.1$ for $h = 5$, and the blue triangles for a maximal error of 10^{-4}, corresponding to $R = 1.8$ rsp. $R = 0.3$. Open symbols show the maximal error that occurred within all particles with similar z–coordinates, the filled symbols show the RMS force error of these particles.

exponentially with decreasing distance to the borders, even in the case where no ELC terms are added. This strongly supports our warning to choose the error bound for ELC rsp. the Yeh and Berkowitz method small enough. For the calculation presented here, at a maximal pairwise error of 10^{-4} the error is smaller than the P3M maximal error.

Fig. 4.4 shows the computation time results for a system as described above with ELC tuned to an maximal pairwise error of 10^{-6}, while P3M is still tuned for a RMS force error of 10^{-4}. With this combination of error bounds, even for 10000 particles no visible error accumulation near the borders occur. Additionally $\delta = 0.2\lambda$ and $\delta = 0.05\lambda$ were used. The computation time for P3M was equal for all settings of δ up to time measurement errors. At 3000 to 5000 particles the L2 cache limit causes a transition to a higher prefactor for ELC similar to MMM2D. At 6000 particles P3M changes the mesh grid size from 32 to 64 mesh points per side, which results in a jump in the computation time. The computation time of MMM2D is considerably higher, so that MMM2D is only useful if excessive precision is needed or for small particle numbers below 200.

The crossover of the computation time between ELC and P3M occurs for δ around 0.2λ. A computation using the plain Yeh and Berkowitz method requires a simulation box with $\lambda_z = 4\lambda$ or larger for the same precision. As was argued before, the computation time for P3M scales linearly with λ_z, so that the computation time for a plain Yeh and Berkowitz type calculation can be estimated by multiplying the computation times of P3m as given in Fig. 4.4 by a factor of four. For $h = 0.5\lambda$, i. e. a simulation box filled with particles up to a height half as large as the other box sides, ELC is around $5 - 7$ times faster than P3M, leading to an overall computation time around

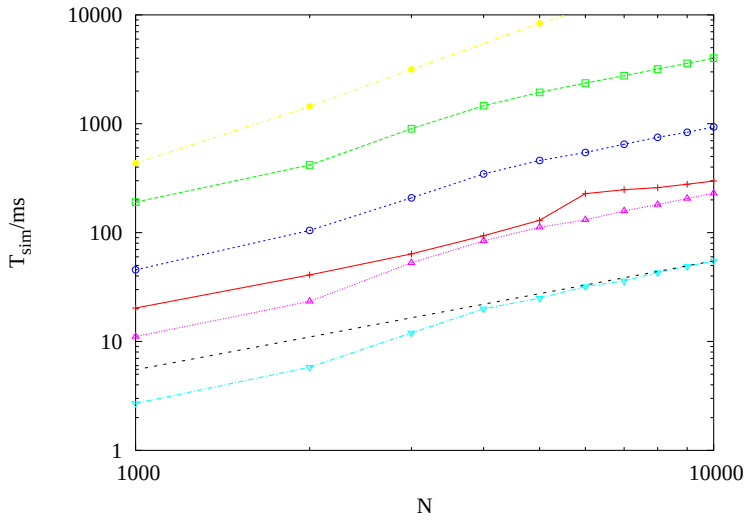


Figure 4.4: Computation times of ELC for systems of N particles and different gap sizes $\delta = 0.5\lambda$ (cyan triangle downs), $\delta = 0.2\lambda$ (violet triangle ups), $\delta = 0.1\lambda$ (blue circles) and the $\delta = 0.05\lambda$ (green rectangles). The red plusses show the computation time of P3M, the yellow filled circles show the timings of MMM2D. The stippled line is a linear fit to the $\delta = 0.5\lambda$ curve.

6/5 of the computation times of P3M, which is more than three times faster than the estimated computation time of the Yeh and Berkowitz method.

Due to the fact that our current implementation of P3M only allows for a cubic simulation box, the fraction λ/λ_z will always be one, which results in a large error for all possible values of the slab height. Even for a system with particles only in a slab of height $h = 0.1\lambda$ simulations the error of the energy may be larger than the energy itself. Therefore using a method for three dimensionally periodic boundary conditions in two dimensional periodicity requires the use of ELC in ESPResSo, plain Yeh and Berkowitz calculations are not possible with reasonable error bounds. But also if one has an implementation of a method for three dimensional periodicity which can handle arbitrary box dimensions, the combination together with ELC should be faster than a method using the plain Yeh and Berkowitz approach, since the theoretical prediction for the optimal gap size is constant, so that the optimal cutoff for the ELC term will always be quite different from zero. This in turn means that the plain Yeh and Berkowitz method is never optimal.

5 Different dielectric constants

A special problem in the calculation of the electrostatic interaction arises from the fact that in the non periodic direction one often has materials with different dielectric constants inside the simulation space and outside. For example water has a dielectric constant around $\epsilon = 80$, while vacuum has $\epsilon = 1$. Therefore a thin film of water containing charged particles will have non–negligible polarisation effects. Similar effects occur at a metallic surface, such as found in catalysators.

Technically, one has to deal with the situation were on either one or both sides of the simulation box in the nonperiodic direction the dielectric constant is different from the one inside the simulation box (see Fig. 5.1). The resulting polarisation effects can be handled using image charges. The following two sections show briefly how this can be handled by MMM2D rsp. ELC, following the approach presented by Smith for the method of Nijboer and de Wette [88].

5.1 Single surface

In the case of a single surface the image charges are just a mirrored picture of the charges in the simulation box itself, where every charge is multiplied with the factor

$$\Delta = \frac{\epsilon_o - \epsilon_i}{\epsilon_o + \epsilon_i} = 1 - 2\frac{1}{1 + \frac{\epsilon_o}{\epsilon_i}} \in [0, 1]\,. \tag{5.1}$$

ϵ_i denotes the dielectric constant in the simulation region, ϵ_o denotes the dielectric constant outside, i. e. inside the wall. If the wall is metallic as in the first example above, $\epsilon_o = \infty$ and $\Delta = 1$, i. e. the image charges have full charge. These image charges are treated identically to the image charges in the periodic direction in that they only contribute through their potential on the charges inside the simulation box, but do not interact with other image charges.

With MMM2D the treatment of this case is fairly easy. Assume that the surface is the bottom border of the simulation box with a z–coordinate of $z = 0$, as shown on the right of Fig. 5.1. Except for the lowest layer, the image particles are all sufficiently

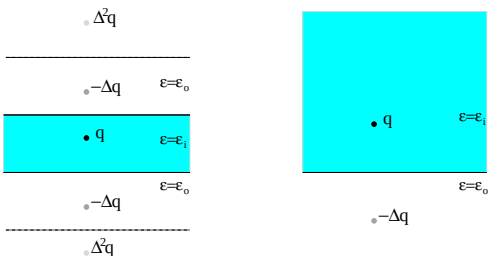


Figure 5.1: Schematic representation of the two systems with different dielectric constants as described in the text, on the left with two interfaces, on the right with only one. In the cyan regions $\epsilon = \epsilon_i$, in the white regions $\epsilon = \epsilon_o$. The solid lines represent the surfaces, the dashed lines are periodic replications of the surfaces.

far away from the layers of the simulation box that their interaction can be calculated using the far formula. This means that they occur only as additional contributions to the $\Xi_s^{(l,s/c,s/c)}$ as defined in Eqn. (3.30). Since the mirroring just exchanges the signs of the z–coordinates of the particles, the additional contribution is given by $\Xi_{B+1}^{(h,s/c,s/c)}$, which the top processor node can calculate easily. This value is then broadcasted to all other nodes and added to all $\Xi_s^{(l,s/c,s/c)}$. Since some of the particles in the lowest layer may be closer to their images then allowed for the far formula, the interaction between the lowest layer has be calculated using the near formula. Therefore for this layer $\Xi_0^{(-,s/c,s/c)}$ has to be subtracted from $\Xi_0^{(l,s/c,s/c)}$ again, so that these interactions are not treated by the far formula, and the interactions of the particles of the lowest layer with their mirrored images have to be calculated by the near formula explicitly.

With ELC the treatment of this case is in general difficult. The additional image charges cannot be treated like real charges since then their interactions with each other will be calculated as well. This is fine if one is interested in the force, but for the energy there is no simple way of subtracting the unwanted contribution of the images charges again. ELC can only be used if the real charges are kept away from the surface by some distance, for example by their finite volume. In this case the interactions with the images can be treated by the far formula alone (of course this also holds for MMM2D, but is not necessary there). In typical simulations the minimal distance to the border will be around 1/100 of the other dimensions, so that the calculation of the image charges will need a considerable amount of computation time. A calculation using the plain Yeh and Berkowitz approach is not possible, since from the image particles some interactions have to be taken into account and some not. This cannot be achieved with e. g. the Ewald method, so that ELC is still the best approach for large numbers of particles.

5.2 Two surfaces — thin films

The case of a two surfaces, i. e. a thin film, is more complex. The reason is that not only the particles in the primary simulation box generate image charges, but also the image charges themselves at the opposite surface. This leads to an infinite array of charges with exponentially decaying charge, as shown in Fig. 5.1. We assume that the surface are at $z = 0$ and $z = \lambda_z$. The image charges of a single charge q_i located at p_i are located at

$$\begin{aligned} p_{2m-1} &= -p_i + n_{0,0,2m}\,, \quad q_{2m-1} = -\Delta^{|2m-1|} q_i \quad \text{and} \\ p_{2m} &= p_i + n_{0,0,2m}\,, \quad q_{2m} = \Delta^{|2m|} q_i \quad \forall n \in \mathbb{Z}\,. \end{aligned} \tag{5.2}$$

If $\Delta = 1$, i. e. if metallic boundary conditions are applied, the sum equals to the standard sum with three dimensional periodic boundary conditions, except for the alternating signs. By expanding the system to contain one of the generated images charges, i. e. double λ_z, one can use any method for three dimensional periodicity. The order of summation does not matter, which is reflected by the fact that the resulting system necessarily has a net dipole moment of 0.

If $\Delta < 1$, the problem is very similar to ELC. The prefactors $\Delta^{2|m|}$ rsp. $\Delta^{|2m-1|}$ can be treated easily together with the factors $e^{2\pi f_{pq}|m|\lambda_z}$ that occur in Eqn. (4.10).

If we again assume that the particles are kept sufficiently far away from the border, the calculation is identical to the ELC algorithm except for some prefactors, which we will now derive. We start with $m \geq 1$, i. e. layers above the primary simulation box. We obtain in analogy to Eqn.(4.10)

$$-\sum_{m\geq 1} \Delta^{|2m-1|} e^{2\pi f_{pq}|-z_i-z_j+2m\lambda_z|} = -\frac{e^{2\pi f_{pq}(-z_i-z_j)}}{\Delta\left(\Delta^2 e^{4\pi f_{pq}\lambda_z}-1\right)} \tag{5.3}$$

and

$$\sum_{m\geq 1} \Delta^{|2m|} e^{2\pi f_{pq}|z_i-z_j+2m\lambda_z|} = \frac{e^{2\pi f_{pq}(z_i-z_j)}}{\Delta^2 e^{4\pi f_{pq}\lambda_z}-1}\,. \tag{5.4}$$

For $m \leq -1$ we obtain

$$-\sum_{m\leq -1} \Delta^{|2m-1|} e^{2\pi f_{pq}|-z_i-z_j+2m\lambda_z|} = -\frac{\Delta e^{2\pi f_{pq}(z_i+z_j)}}{\Delta^2 e^{4\pi f_{pq}\lambda_z}-1} \tag{5.5}$$

and

$$\sum_{m\leq -1} \Delta^{|2m|} e^{2\pi f_{pq}|z_i-z_j+2m\lambda_z|} = \frac{e^{2\pi f_{pq}(-z_i+z_j)}}{\Delta^2 e^{4\pi f_{pq}\lambda_z}-1}\,. \tag{5.6}$$

Now not only $z_i - z_j$ occurs in the exponents, but also $z_i + z_j$, and the prefactors are different than in Eqn. (4.10), but the resulting formula can be treated algorithmically identical as for ELC with the same computational efficiency.

If the particles can get arbitrarily close to the surfaces, only MMM2D can be used. This time for every layer s one needs all sums $\sum_{t<s-1} \Xi_t^{(\pm,s/c,s/c)}$, $\sum_{s-1\leq t\leq s+1} \Xi_t^{(\pm,s/c,s/c)}$ and $\sum_{t>s+1} \Xi_t^{(\pm,s/c,s/c)}$ instead of only $\Xi_s^{(l,s/c,s/c)} = \sum_{t<s-1} \Xi_t^{(+,s/c,s/c)}$ and $\Xi_s^{(h,s/c,s/c)} = \sum_{t>s+1} \Xi_t^{(-,s/c,s/c)}$ as needed for the plain MMM2D method. They can be obtained in the same sequential communication as described for MMM2D, just the data amount doubles, since now both the terms with positive and negative sign in the exponential have to be communicated up and down. The prefactors are determined by adding up the prefactors obtained from the far formula of MMM2D, which are just ± 1 or 0, and the prefactors obtained from Eqns. (5.3) through (5.6), which are of the form $\pm\Delta^\nu \left(e^{4\pi f_{pq}\lambda_z}-1\right)^{-1}$. The signs of the prefactors have to be obtained from the addition theorems for sine and cosine.

Similar to the single surface case, for the lowest layer $s = 0$ and the highest layer $s = B - 1$ the prefactors are different than for the other layers, since the interaction with their own mirrored image layers has to be done using the near formula, as particles can be arbitrarily close to their images (and probably will be, as the image charges have opposite sign). In this case, the sum over $m \geq 1$ rsp. $m \leq -1$ of the odd terms turns into a sum over $m \geq 2$ rsp. $m \leq -2$, raising an additional prefactor of $\Delta^{-2} e^{4\pi f_{pq}\lambda_z}$ rsp. $\Delta^2 e^{4\pi f_{pq}\lambda_z}$.

The computation time of both MMM2D rsp. ELC will increase considerably if the algorithms given above for treatment of different dielectric constants are implemented. Nevertheless the algorithmic details remain, therefore the overall computation time scaling is still valid, i. e. $\mathcal{O}(N^{5/3})$ for MMM2D and $\mathcal{O}(N)$ for the ELC term. Moreover due to the different approach, an adaption of the Ewald type methods especially to the

case of two surfaces cannot be done as easily. For these methods, the contributions of the images cannot be calculated analytically, and consequently no adaption of these methods for the case different dielectrica has been reported so far.

6 One dimensional periodicity

After treating the case of two dimensionally periodically replicated systems it is only naturally to turn to the case of one dimensionally periodic systems. Such a periodicity is useful e. g. in researching properties of long stiff polymers such as DNA or nano–pores. The DNA is modelled as an infinitely long rod or helix, which is obtained again by periodic replication. This will be illuminated in the following chapter. In general a one dimensionally periodic system is suitable if the problem in question is locally translationally invariant in exactly one of the coordinates.

So far only a few methods for this periodicity are known. Besides the multipole methods for large numbers of particles and the Lekner method, which are suitable for all kinds of boundary conditions, the only method specifically aimed at systems with one dimensional periodicity is the $1d$–Ewald sum. This method has the same two drawbacks as the $2d$–Ewald method: it is computationally slow to evaluate and no accurate error estimates exist.

The formulas from MMM2D can easily be reused for one dimensional periodicity. In this case, not even the algorithmic tricks of MMM2D cannot be applied, since the obtained far formula does not allow a simple product decomposition at all. Therefore MMM1D has a complexity of $\mathcal{O}(N^2)$, but small problems can still be tackled since the prefactor is small. For MMM1D rigorous error bounds similar to MMM2D exist. The unfavourable scaling of the methods for this periodicity is remedied by the fact that the number of particles needed to avoid finite size effects is much smaller than for three dimensionally periodicity; most simulations can be performed with less than 500 particles.

The first section of this chapter will present a proof that also in one–dimensionally periodic systems the convergence factor approach is equivalent to the spherical summation approach. In the following sections the MMM1D formulas are derived and the corresponding error formulas, and finally some timing results are given, showing that the method can be used efficiently for up to 300 particles on a single processor.

6.1 General prerequisites

Without loss of generality, we assume that the periodic coordinate is z. The particles are assumed to reside in a primary simulation box of dimensions

$$B_0 = \left(-\infty, \infty\right) \times \left(-\infty, \infty\right) \times \left(-\frac{\lambda_z}{2}, \frac{\lambda_z}{2}\right]. \tag{6.1}$$

Since now the only periodic coordinate is z, the notations differ slightly from the notations given in Sec. 2.1, as used in chapters 2 and 3. Here we use the notation

$$\rho := \sqrt{x^2 + y^2} \quad \text{rsp.} \quad \rho_{ij} := \sqrt{x_{ij}^2 + y_{ij}^2} \tag{6.2}$$

for the projection of the (particle) coordinates onto the (x, y)–plane.

$$n_k = (0, 0, k\lambda_z) \tag{6.3}$$

denotes the base vector of the k–th periodic image, and

$$r_k := |r + n_k| = \sqrt{\rho^2 + (z + k\lambda_z)^2} \tag{6.4}$$

the distance from the origin to the k–th periodic image of position r.

Using this notations, we can write down the Coulomb energy of a system with periodicity only along the z–axis as

$$E = \frac{1}{2} \sum_{S=0}^{\infty} \sum_{k=\pm S} \sum_{i,j}{}' \frac{q_i q_j}{|p_{ij} + n_k|} \,. \tag{6.5}$$

Again, the order is spherical in the hypergeometric sense, although this order this time reduces to adding up the contributions symmetrically. Using the same argument as given in chapter 3 one can see that the Coulomb sum in a one dimensionally system is absolutely convergent as long as the contributions are added symmetrically. Therefore again no shape–dependent term occurs.

6.2 1d–Ewald method

The Ewald method for a one dimensionally periodic system was described first by M. Porto [81]. Similar to the two dimensional periodicity case, the formula does not allow a product decomposition and therefore has to be calculated with an unfavourable $\mathcal{O}(N^2)$ loop. Moreover no accurate error bounds have been reported so far. The energy is calculated as follows:

$$\begin{aligned} E =& \frac{1}{2} \sum_{i,j} q_i q_j \left(\sum_k{}' \frac{\operatorname{erfc}(\alpha |r_{ij} + n_k|)}{|r_{ij} + n_k|} + \frac{1}{2\lambda_z} \sum_{m \neq 0} G_m(\rho_{ij}, \alpha) e^{\mathrm{i} m z_{ij}} \right) + \\ & \frac{1}{2\lambda_z} \sum_{\substack{i,j \\ \rho_{ij} \neq 0}} q_i q_j \left(-\gamma - \Gamma(0, \alpha^2(\rho_{ij}^2)) - \log(\alpha^2(\rho_{ij}^2)) \right) + \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2 \,, \end{aligned} \tag{6.6}$$

where $\Gamma(n, x)$ denotes the incomplete gamma function [1] and γ Euler's constant. The coefficients of the Fourier transformation are given by

$$G_m(\rho, \alpha) = \begin{cases} \sum_{k \geq 0} \frac{(-1)^k m^{2k}}{4^k k!} \rho^{2k} \Gamma\left(-k, \frac{m^2}{4\alpha^2}\right) & \text{for } \rho \neq 0 \\ \Gamma\left(0, \frac{m^2}{4\alpha^2}\right) & \text{for } \rho = 0 \end{cases} .$$

The additional sum of k inside the $G_m(\rho, \alpha)$ generates an additional cutoff besides the Fourier and real space cutoffs with which one has to deal. Since no error estimates are known, using of this method requires extensive numerical tests to tune α and the cutoffs. Moreover typically 100-200 terms of the Fourier space sum have to be included, which makes the method quite slow because the evaluation of the incomplete gamma function in software is slow.

6.3 Equivalence of the convergence factor approach and the spherical summation

Like in MMM and MMM2D, for MMM1D the Coulomb sum is modified by the application of a convergence factor $e^{-\beta r}$, i. e. the image contribution to the potential reads

$$\tilde{\phi}_\beta(x, y, z) = \sum_{k \neq 0} \frac{e^{-\beta r_k}}{r_k}\,, \tag{6.7}$$

while the total energy $\tilde{E}$ is defined identically to the three dimensional case, see Eqn. (2.20). Similar to the two dimensionally periodic case it can be proven that $E = \tilde{E}$, i. e. that the convergence factor is equivalent to the generalised spherical summation order. To this aim, we look at the difference between E and $\tilde{E}$ as given by

$$2(\tilde{E} - E) = \sum_{k>0} \sum_{i,j} q_i q_j \left(\frac{e^{-\beta|r_k|} - 1}{|r_k|} + \frac{e^{-\beta|r_{-k}|} - 1}{|r_{-k}|} \right). \tag{6.8}$$

To show that this difference is zero, we insert the terms $e^{-\beta|n_k|}/|r_{\pm k}|$, i. e. we introduce a different convergence factor $e^{-\beta|n_k|}$. Then the difference is given by

$$\begin{aligned} 2(\tilde{E} - E) = &\sum_{k>0} \sum_{i,j} q_i q_j \left(\frac{e^{-\beta|r_k|} - e^{-\beta|n_k|}}{|r_k|} + \frac{e^{-\beta|r_{-k}|} - e^{-\beta|n_k|}}{|r_{-k}|} \right) + \\ &\sum_{k>0} \sum_{i,j} q_i q_j \left(\frac{e^{-\beta|n_k|} - 1}{|r_k|} + \frac{e^{-\beta|n_{-k}|} - 1}{|r_{-k}|} \right). \end{aligned} \tag{6.9}$$

We show that both sums vanish in the limit $\beta \to 0$, i. e. that the convergence factors $e^{-\beta|r_{ij}+n_k|}$ and $e^{-\beta|n_k|}$ are equivalent and that the latter one is equivalent to the spherical summation order in turn.

Although the difference between $e^{-\beta|r_k|}$ and $e^{-\beta|n_k|}$ is small, Smith showed that the convergence factor $e^{-\beta|n_k|}$ leads to the same result as a spherical summation even with three dimensional periodicity, in contrast to the factor $e^{-\beta|r_k|}$ [87]. For the case of one and two dimensional periodicity it is sufficient to know that $1/|r + n_k| + 1/|r - n_k| - 2/|n_k| = \mathcal{O}(|n_k|^3)$. Then because of charge neutrality, we can add the $-2/|n_k|$

term needed inside the i,j–sum without changing the value and obtain

$$\begin{aligned}
&\sum_{k>0}\sum_{i,j} q_i q_j \left(\frac{e^{-\beta|n_k|}-1}{|r_k|}+\frac{e^{-\beta|n_k|}-1}{|r_{-k}|}\right)\\
=&\sum_{k>0}\left(e^{-\beta|n_k|}-1\right)\sum_{i,j} q_i q_j \left(\frac{1}{|r_k|}+\frac{1}{|r_{-k}|}-2\frac{1}{|r_k|}\right)\\
=&\sum_{k>0}\left(e^{-\beta|n_k|}-1\right)\mathcal{O}\left(|n_k|^{-3}\right)=\mathcal{O}(\beta)\,,
\end{aligned} \tag{6.10}$$

which shows that the second sum in Eqn. (6.9) vanishes in the limit $\beta \to 0$.

However the first sum of Eqn. (6.9) is more delicate. For $\beta|r_k - n_k| < 1$

$$\begin{aligned}
\frac{e^{-\beta(|r_{\pm k}|-|n_k|)}-1}{|r_{\pm k}|} &= \frac{1}{|r_{\pm k}|}\sum_{l=1}^{\infty}\frac{(-1)^l}{l!}\beta^l(|r_{\pm k}|-|n_k|)^l\\
&= \beta\left(\frac{|n_k|}{|r_{\pm k}|}-1\right)+\theta(\beta,|r_{\pm k}|,|n_k|)\,,
\end{aligned} \tag{6.11}$$

where $|\theta(\beta,|r_{\pm k}|,|n_k|)| \le \frac{1}{2}\frac{\beta^2(|r_{\pm k}|-|n_k|)^2}{|r_{\pm k}|}$. Inserting this into the first sum of Eqn. (6.9) leads to

$$\begin{aligned}
&\left|\sum_{k>0} e^{-\beta|n_k|}\sum_{i,j} q_i q_j \left(\frac{e^{-\beta(|r_k|-|n_k|)}-1}{|r_k|}+\frac{e^{-\beta(|r_{-k}|-|n_k|)}-1}{|r_{-k}|}\right)\right|\\
\le&\sum_{k>0} e^{-\beta|n_k|}\sum_{i,j}|q_i q_j|\left|\frac{\beta}{r_k}\left(\frac{|n_k|}{|r_k|}-1\right)+\frac{\beta}{r_{-k}}\left(\frac{|n_k|}{|r_{-k}|}-1\right)+\right.\\
&\qquad\left.\theta(\beta,|r_k|,|n_k|)+\theta(\beta,|r_{-k}|,|n_k|)\right|\\
\le&\sum_{k>0} e^{-\beta|n_k|}\sum_{i,j}|q_i q_j|\mathcal{O}\left(\frac{\beta}{|n_k|^2}\right)+\sum_{k>0} e^{-\beta|n_k|}\sum_{i,j}|q_i q_j|\mathcal{O}\left(\frac{\beta^2}{|n_k|}\right)=\mathcal{O}(\beta)\,.
\end{aligned} \tag{6.12}$$

This argument still holds if the summation over k is replaced by a two dimensional lattice, but for the three dimensional lattice the argument is not valid.

6.4 MMM1D

For $\rho > 0$ and $\beta > 0$, we can Fourier transform the sum in Eqn. (6.7) to a sum over Bessel functions as was also done in the near formula (3.11) of MMM2D. This gives

the far formula

$$\begin{aligned}\phi_\beta(\rho,z) &= \sum_{k\in\mathbb{Z}}\frac{e^{-\beta r_k}}{r_k}\\ &= 2u_z\sum_{p\in\mathbb{Z}}\mathrm{K}_0(\sqrt{\beta^2+(2\pi u_z p)^2}\rho)e^{2\pi i u_z p z}\\ &= 4u_z\sum_{p\neq 0}\mathrm{K}_0(2\pi u_z p\rho)\cos(2\pi u_z p z)-\\ &\quad 2u_z\log(\lambda_z\beta)-2u_z\log(\rho/(2\lambda_z))-2u_z\gamma+\mathcal{O}(\beta)\,.\end{aligned}$$

For $\rho\leq 1/2\lambda_z$, $u_z\rho < N_\psi - 1/2$ we can use

$$\begin{aligned}\tilde{\phi}_\beta(\rho,z) &= \sum_{k\neq 0}\frac{e^{-\beta r_k}}{r_k}\\ &= -u_z\sum_{n\geq 0}\binom{-\frac{1}{2}}{n}\frac{\left(\psi^{(2n)}(N_\psi+u_z z)+\psi^{(2n)}(N_\psi-u_z z)\right)}{(2n)!}(u_z\rho)^{2n}-\\ &\quad 2u_z\gamma-2u_z\log(\lambda_z\beta)+\sum_{k=1}^{N_\psi-1}\left(\frac{1}{r_k}+\frac{1}{r_{-k}}\right)+\mathcal{O}(\beta))\,,\end{aligned}$$

which is the same formula as used in the one dimensional sum occurring in the near formula of MMM2D. The tilde again denotes the regularised potential omitting the central box interaction.

The self energy for a particle is

$$\sum_{k\neq 0}\frac{e^{-\beta|k\lambda_z|}}{|k\lambda_z|} = -2u_z\log(1-e^{-\beta\lambda_z}) = -2u_z\log(\lambda_z\beta)+\mathcal{O}(\beta)\,,$$

i. e. there is no contribution except the singularity and the self energy will not occur in the numerical evaluation.

We define $\omega = 2\pi u_z p$. For $\rho > 0$ the *far formula in one dimension* is given by

$$\begin{aligned}\phi(\rho,z) &= 4u_z\sum_{p\neq 0}\mathrm{K}_0(\omega\rho)\cos(\omega z)-2u_z\log\left(\frac{\rho}{2\lambda_z}\right)-2u_z\gamma\\ F_\rho(\rho,z) &= 8\pi u_z^2\sum_{p\neq 0}p\,\mathrm{K}_1(\omega\rho)\cos(\omega z)+\frac{2u_z}{\rho}\\ F_z(\rho,z) &= 8\pi u_z^2\sum_{p\neq 0}p\,\mathrm{K}_0(\omega\rho)\sin(\omega z)\,.\end{aligned}\tag{6.13}$$

From ϕ the energy is obtained as

$$E = \frac{1}{2}\sum_{i,j}q_i q_j\phi(\rho_{ij},z_{ij})\,.\tag{6.14}$$

Similarly the force exerted on particle i is given by

$$F_i = q_i\sum_j q_j F(\rho_{ij},z_{ij})\tag{6.15}$$

where $F(\rho,z)=(F_\rho(\rho,z)x/\rho, F_\rho(\rho,z)y/\rho, F_z(\rho,z))$.

For small ρ we obtain the *near formula in one dimension*

$$\begin{aligned}
\tilde{\phi}(\rho,z) =& -u_z\sum_{n\geq 0}\binom{-\frac{1}{2}}{n}\frac{\left(\psi^{(2n)}(N_\psi+u_z z)+\psi^{(2n)}(N_\psi-u_z z)\right)}{(2n)!}(u_z\rho)^{2n}-2u_z\gamma+\\
&\sum_{k=1}^{N_\psi-1}\left(\frac{1}{r_k}+\frac{1}{r_{-k}}\right)\\
\tilde{F}_\rho(\rho,z) =& -u_z^3\sum_{n\geq 0}\binom{-\frac{1}{2}}{n}\frac{\left(\psi^{(2n)}(N_\psi+u_z z)+\psi^{(2n)}(N_\psi-u_z z)\right)}{(2n)!}(u_z\rho)^{2n-1}+\\
&\sum_{k=1}^{N_\psi-1}\left(\frac{\rho}{r_k^3}+\frac{\rho}{r_{-k}^3}\right)\\
\tilde{F}_z(\rho,z) =& -u_z^2\sum_{n\geq 0}\binom{-\frac{1}{2}}{n}\frac{\left(\psi^{(2n+1)}(N_\psi+u_z z)+\psi^{(2n+1)}(N_\psi-u_z z)\right)}{(2n)!}(u_z\rho)^{2n}+\\
&\sum_{k=1}^{N_\psi-1}\left(\frac{z+k\lambda_z}{r_k^3}+\frac{z-k\lambda_z}{r_{-k}^3}\right),
\end{aligned} \tag{6.16}$$

where the tilde denotes that the interactions in the primary simulation box are left out and have to be added separately. Otherwise the expressions of the near and far formula can be exchanged freely.

6.5 Error estimates

An upper bound on the absolute errors of the energy of the far formula for the summation up to but not including the finite cutoff P is

$$|\phi(\rho,z)-\phi(P,\rho,z)| \leq 4u_z\,\mathrm{K}_0(2\pi u_z P\rho_{min})\frac{e^{2\pi u_z\rho_{min}}}{2\pi u_z\rho_{min}}$$

rsp. for the forces

$$|F_\rho(\rho,z)-F_\rho(P,\rho,z)|_\infty \leq 8\pi u_z^2\,\mathrm{K}_1(2\pi u_z P\rho_{min})\frac{e^{2\pi u_z\rho_{min}}}{2\pi u_z\rho_{min}}\left(P-1+\frac{1}{2\pi u_z\rho_{min}}\right)$$

provided that $\rho\geq\rho_{min}$. This formulas are obtained similarly to Eqn.(3.21). ρ_{min} will in general be the radius where the calculation switches from the far to the near formula.

The near formula can be treated by the Leibniz criterion as usual and no further error analysis is needed. Note that for the simplicity of the implementation one will normally choose a fixed N_ψ. Moreover for $(u_z\rho)/(N_\psi-1/2)\to 1$ the convergence of the polygamma series will become poor. Therefore one will check that $\rho<\theta\lambda_z(N_\psi-1/2)$ where $0<\theta<1$ is a constant which determines the number of polygamma terms calculated. The larger θ is, the larger ρ may be, but more polygamma terms have to be calculated in the worst case. In the implementation of MMM2D N_ψ is fixed

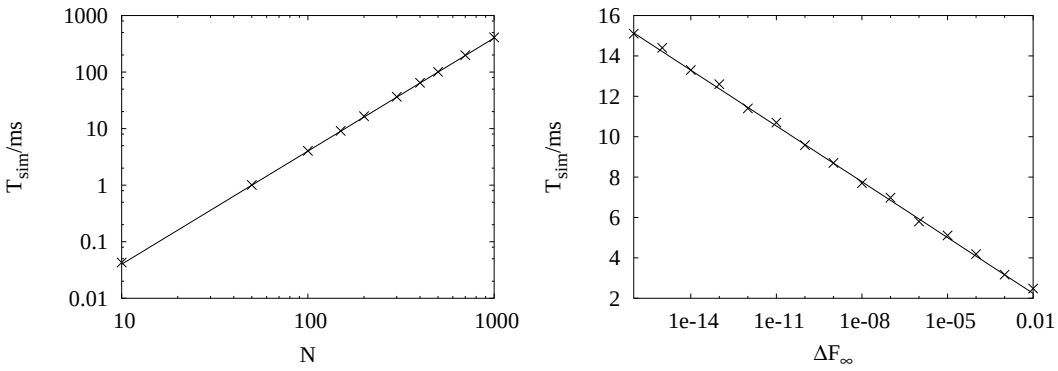


Figure 6.1: Computation time of MMM1D in milliseconds for varying number of particles N (left graph) and varying force error bounds ΔF_∞ (right graph). The straight line gives a fit of $\mathcal{O}(N^2)$ (left graph) rsp. $-log(\epsilon)$ to emphasise the scaling.

to a value of 2 to avoid the additional summation of direct interactions. Numerical tests show that the optimal computation time is achieved for $\rho << \lambda_z$. Therefore in ESPResSo θ is fixed to 2/3, which allows ρ to take any value between 0 and λ_z.

Because the error estimates are the same as for MMM, MMM2D and ELC, it is unnecessary to show the error dependencies here again. The same warning as for the other methods holds: the error drops exponentially outside the switching radius. On the other hand since the formulas are radially invariant, the radial error distribution is uniform.

Although MMM1D has a clearly unfavourable scaling of $\mathcal{O}(N^2)$, it still can be used effectively in a simulation, especially because of its small prefactors. The evaluation of the incomplete gamma function and the Bessel function consume a similar amount of computation time, but instead of 100-200 Fourier space terms, the far formula achieves good precision with less than 20 terms of the Bessel sum.

The left graph in Fig. 6.1 shows the computation time of the implementation of MMM1D in ESPResSo for the force calculation. The test system consists of randomly positioned unit charges in a simulation box of fixed size $10 \times 10 \times 10$, MMM1D was tuned to a maximal pairwise force error of 10^{-4}. The computations were performed on a single AMD Athlon64 3000+ processor. One can see clearly the expected computation time scaling of $\mathcal{O}(N^2)$. The second graph in Fig. 6.1 shows the computation time for 100 particles at various error bounds. The scaling is logarithmic, similar to MMM2D(see Fig. 3.8) or MMM. Using again the criterion of less than $0.1s$ per time step, MMM1D can be used efficiently for up to 400 particles.

6.6 Formulas for rods

For the problem we want to investigate in the next chapter we need the Coulombic forces exerted by a charged rod both on a particle and on another charged rod. To be able to combine these potentials with the MMM1D method, they have to be calculated using the same convergence factor. Especially it is important to see that the rod potential generates the same singularity in β, so that for a system of charges and rods the singularity still vanishes if the overall system is charge neutral.

The energy between a rod of line charge density τ and a particle of charge q is

$$\begin{aligned}
&\int_0^{\lambda_z} \sum_{k\in\mathbb{Z}} q\tau\phi_\beta(\rho, z-\tilde{z})d\tilde{z} \\
&= \int_0^{\lambda_z} q\tau\Bigg(4u_z \sum_{p\neq 0} \mathrm{K}_0(2\pi u_z p\rho)\cos(2\pi u_z p(z-\tilde{z})) \\
&\qquad - 2u_z\log(\lambda_z\beta) - 2u_z\log(\rho/(2\lambda_z)) - 2u_z\gamma\Bigg)d\tilde{z} + \mathcal{O}(\beta) \\
&= -2q\tau\left(\log(\lambda_z\beta) + \log\left(\rho/(2\lambda_z)\right) + \gamma\right) + \mathcal{O}(\beta)\,,
\end{aligned} \tag{6.17}$$

which recovers the classical textbook result up to the constant term $q\tau\gamma$ and the singularity. Note that the charge of the rod section inside the simulation box is $Q = \tau\lambda_z$, so that the singularity in Eqn. (6.17) is $-2u_z qQ\log(\lambda_z\beta)$, which is consistent with the singularity found in the particle–particle energy formulas.

For two rods of line charges τ and τ' the interaction energy is therefore

$$-2\tau\tau'\lambda_z\left(\log(\lambda_z\beta) + \log\left(\rho/(2\lambda_z)\right) + \gamma\right) + \mathcal{O}(\beta)\,. \tag{6.18}$$

As one can see clearly, the rod–rod energy diverges as the rods get close, especially the self energy of a rod is infinite has to be omitted from any energy calculation. Similar to the rod–particle energy, the singularity has a form consistent with the singularity of the MMM1D particle–particle energy formulas, so that the singularity vanishes in a charge neutral system consisting of charged particles and rods.

As usually, we finally give the energies and forces after the limit $\beta \to 0$. For the interaction between a rod and a particle we have

$$\begin{aligned}
E^{rod}(\rho, z) &= -2q\tau\log\left(\rho/(2\lambda_z)\right) - 2q\tau\gamma \\
F_\rho^{rod}(\rho, z) &= 2q\tau/\rho \\
F_z^{rod}(\rho, z) &= 0
\end{aligned} \tag{6.19}$$

and for the interaction of two rods

$$\begin{aligned}
E^{rod,rod}(\rho) &= -2\tau\tau'\lambda_z\log\left(\rho/(2\lambda_z)\right) - 2\tau\tau'\lambda_z\gamma \\
F_\rho^{rod,rod}(\rho) &= 2\tau\tau'\lambda_z/\rho\,.
\end{aligned} \tag{6.20}$$

7 Applications of MMM1D: The two rod system

In a polar solvent, macromolecules possessing dissociable groups can acquire a charge by dissociation like a normal salt. These charged polymers, called polyelectrolytes, or charged colloids are then surrounded by a layer of oppositely charged counterions. It is well known that sufficiently strong electrostatic interactions lead to behaviour which cannot be described on the mean-field level. It can be proven rigorously that the standard mean field Poisson–Boltzmann theory will predict repulsion between like–charged macroions, regardless of their shape [72,95,94,85]. But attractive interactions and other non–mean–field phenomena like overcharging have been confirmed by a large number of computer simulations [40, 41, 58, 59, 39, 22, 53]. Experiments have shown that DNA, a stiff, highly negatively charged polyelectrolyte, can be condensed by multivalent counterions [101, 80, 12]. This correlation–induced attraction is for instance believed to be important for the compaction of DNA inside viral capsids [50, 33]. In this chapter we deal with the case of two infinitely long, charged rigid rods, which can be regarded as simple DNA models or more generally as a good approximation for studying the ion distribution around semi–flexible polyelectrolytes.

The experimental as well as the theoretical observations indicate that attraction of like–charged objects occurs in highly charged systems, i. e. when multivalent counterions are present, the macroions are highly charged, or if the strength of the electrostatic interaction is increased, for example at low temperatures. Often attractive interactions go along with a strong correlation hole in the radial distribution function of the counterions and interlocking patterns along the parallel rods [18]. Several attempts have been made to go beyond the Poisson–Boltzmann mean field theory, using integral equations [34, 23], density functional theories [10], field theoretical calculations [65, 67, 66] or other approaches [42, 86, 49]. For a general overview on the topic see the articles of Deserno, Jönsson, Kjellander and Netz in [44].

In Sec. 7.1, the classical Poisson–Boltzmann theory for a single rod and the concept of Manning condensation are briefly reviewed. Sec. 7.2 presents the strong coupling (SC) theory and its application to the problem of two like–charged rods. In contrast to the Poisson–Boltzmann theory, the strong coupling theory is aimed at the limit of infinite counterion–counterion coupling and is able to predict attraction between the charged rods. Sec. 7.3 compares results for the equilibrium surface to surface distance of two rods and the counterion distribution from computer simulations using MMM1D to the predictions of the strong coupling theory. The numerical data fits well to the SC prediction, although the simulations are performed at only moderately high coupling. It is shown that the Rouzina–Bloomfield parameter γ_{RB} can be used to quantify the degree of agreement between the simulation and the theory.

In Sec. 7.4 the problem is viewed in the special case of zero temperature. This limit is identical to the limit of infinite Bjerrum length, and therefore infinite counterion–

counterion coupling, compared to the thermal energy. The SC theory predicts attraction of the two like–charged rods for all distances, so that their equilibrium position is side by side. The counterions are predicted to line up in the gap between the two rods. Some simple analytical calculations using the MMM1D formulas will demonstrate that this is not true in general, and a parameter similar to the Rouzina–Bloomfield parameter is identified which determines whether the counterions will accumulate in the gap. Moreover, the counterions are located in the plane spanned by the two rods for a larger range of this parameter, until ultimately they leave the plane. These predictions are confirmed by simulation results presented in Sec. 7.6.

7.1 Poisson–Boltzmann Theory

Consider a system consisting of N counterions of charge qe and mass m, and a fixed charge distribution $\rho_f(r)e$. Note that both the counterion charge and the fixed charge density are given in multiples of the unit charge e. The potential energy of this system is given by

$$\beta U = \sum_{i<j} \frac{q^2 \ell_B}{|r_i - r_j|} + \int_r \sum_i \frac{q\rho_f(r)\ell_B}{|r - r_i|} dr + \frac{1}{2} \int_r \int_{r'} \frac{\rho_f(r)\rho_f(r')\ell_B}{|r - r'|} dr' dr \,, \tag{7.1}$$

where $\beta = 1/(k_B T)$, and ℓ_B is the *Bjerrum length*

$$\ell_B := \frac{e^2}{4\pi\epsilon\epsilon_0 k_B T} \,, \tag{7.2}$$

which measures the distance at which the interaction energy between two unit charges is equal to the thermal energy. Assuming a surface charge density of σ_s for the fixed charge distribution, the interaction of the counterions with the fixed charge distribution can be quantified by the *Gouy–Chapman length* [35]

$$\mu = -\frac{1}{2\pi q \sigma_s \ell_B} \,, \tag{7.3}$$

which measures the distance from a charged plane of charge density σ_s at which a particle of charge q has an effective interaction of $k_B T$ with the plane. Note that due to charge neutrality, either σ_s or q have to be negative. It will prove convenient to rescale the coordinates by μ via $\tilde{r} = r/\mu$, which gives the electrostatic energy of the system the following simple form:

$$\tilde{U} = \beta U = \sum_{i<j} \frac{\Xi}{|\tilde{r}_i - \tilde{r}_j|} - \frac{1}{2\pi} \int_{\tilde{r}} \sum_i \frac{\tilde{\rho}_f(\tilde{r})}{|\tilde{r} - \tilde{r}_i|} d\tilde{r} + \frac{1}{8\pi^2 \Xi} \int_{\tilde{r}} \int_{\tilde{r}'} \frac{\tilde{\rho}_f(\tilde{r})\tilde{\rho}_f(\tilde{r}')}{|\tilde{r} - \tilde{r}'|} d\tilde{r}' d\tilde{r} \,, \tag{7.4}$$

where $\tilde{\rho}_f(r) = \mu\sigma(r)/\sigma_s$ is the rescaled charge density and

$$\Xi := \frac{q^2 \ell_B}{\mu} = -2\pi q^3 \ell_B^2 \sigma_s \,. \tag{7.5}$$

The dimensionless quantity Ξ is called the *coupling parameter*, since it determines the strength of the counterion–counterion interaction [67]. From the rescaled representation of the electrostatic energy $\tilde{U}$ one can see that in the limit of vanishing Ξ

counterion–counterion correlations, which are given by the first term in Eqn. (7.4), only play a minor role, while they dominate for large Ξ. We start with presenting the Poisson–Boltzmann theory, which is the mean field theory appropriate for weak counterion–counterion coupling.

Poisson–Boltzmann theory

The Hamilton operator for this system is given by $H = \sum_i \frac{p_i^2}{2m} + U$, from which the partition function is obtained as

$$Z = \frac{\mu^{3N}}{N!(2\pi\hbar)^{3N}} \int\limits_{\tilde{r}_1,\ldots,\tilde{r}_N} \int\limits_{p_1,\ldots,p_N} e^{-\beta H} dp_1 \cdots dp_N d\tilde{r}_1 \cdots d\tilde{r}_N \,, \tag{7.6}$$

where the coordinates $\tilde{r}_1, \ldots, \tilde{r}_N$ are integrated only over the space accessible to particles. The momentum part factorises nicely into independent contributions of the single particles which can be calculated analytically, and one obtains

$$Z = \frac{1}{N!\tilde{\lambda}_T^{3N}} \int\limits_{\tilde{r}_1,\ldots,\tilde{r}_N} e^{-\tilde{U}} d\tilde{r}_1 \cdots d\tilde{r}_N \,, \tag{7.7}$$

where $\tilde{\lambda}_T = 2\pi\hbar/(\mu\sqrt{2\pi m k_B T})$ is the rescaled thermal wavelength. However, the positional part of the integral cannot be separated into single particle contributions, since the particle coordinates are coupled in $\tilde{U}$. The main idea of the mean–field treatment is to replace the original N–particle distribution function $p_N = e^{-\tilde{U}(\tilde{r}_1,\ldots,\tilde{r}_N)}/(\int e^{-\tilde{U}(\tilde{r}_1,\ldots,\tilde{r}_N)})$ by a product of single particle distribution functions $p_p := p_1(\tilde{r}_1)\cdots p_1(\tilde{r}_N)$, where $p_1(\tilde{r})$ is the one–particle distribution function. $p_1(\tilde{r})$ is the distribution, which is obtained if a particle interacts with the other particles only through their *mean field*, i. e. their equilibrium particle distribution. The product particle distribution allows to treat the positional integrals particle by particle again.

Although the product distribution is different from the exact Boltzmann–distribution in general, it still gives an upper bound to the exact free energy through the *Gibbs–Bogliubov-inequality* [21],

$$F \leq \langle H \rangle_p - TS_p = \mu^{3N} \int\limits_{\tilde{r}_1,\ldots,\tilde{r}_N} pH \; d\tilde{r}_1 \cdots d\tilde{r}_N + k_B T \mu^{3N} \int\limits_{\tilde{r}_1,\ldots,\tilde{r}_N} p \log p \; d\tilde{r}_1 \cdots d\tilde{r}_N \tag{7.8}$$

which holds for any particle distribution function p. Inserting $p = p_p$ gives after some simple calculations the Poisson–Boltzmann free energy functional

$$F_{PB}(\tilde{\rho}) = k_B T \int\limits_{\tilde{r}} \tilde{\rho}(\tilde{r}) \left(\frac{1}{2}\tilde{\psi}_i(\tilde{r}) + \tilde{\psi}_f(\tilde{r}) + \log(\tilde{\rho}(\tilde{r})\tilde{\lambda}_T^3) - 1 \right) d\tilde{r} \,, \tag{7.9}$$

where $\tilde{\rho}(\tilde{r}) = N\mu^3 p_1(r)$ is the rescaled particle density,

$$\tilde{\psi}_i(\tilde{r}) = \int\limits_{\tilde{r}'} \frac{\Xi\tilde{\rho}(\tilde{r}')}{|\tilde{r} - \tilde{r}'|} d\tilde{r}' \tag{7.10}$$

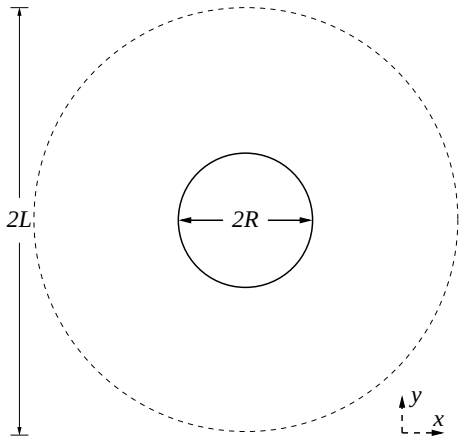


Figure 7.1: Schematic view of the one rod system along the rod axis. The inner circle represents the rod of radius R and line charge density τ, the outer circle a cylindrical compartment of radius L. The particles of charge $-q$ can only access the region enclosed by the two circles.

is the rescaled electrostatic potential due to the particle density $\tilde{\rho}$, and

$$\tilde{\psi}_f(\tilde{r}) = -\frac{1}{2\pi} \int\limits_{\tilde{r}'} \frac{\tilde{\rho}_f(\tilde{r}')}{|\tilde{r} - \tilde{r}'|} d\tilde{r}' \tag{7.11}$$

is the rescaled electrostatic potential due to the fixed charge density $\tilde{\rho}_f$. Note that the constant self energy of the fixed charge distribution is ignored throughout this section. $F_{PB}(\tilde{\rho})$ is an upper bound to the free energy for any normalised p_1 rsp. $\tilde{\rho}$, therefore we are interested in smallest possible upper bound. From a functional minimisation of $F_{PB}(\tilde{\rho})$ with respect to $\tilde{\rho}$ we obtain

$$\tilde{\rho}(\tilde{r}) = \tilde{\rho}_0 e^{-\tilde{\psi}(\tilde{r})} , \tag{7.12}$$

where $\tilde{\psi}(\tilde{r}) = \tilde{\psi}_i(\tilde{r}) + \tilde{\psi}_f(\tilde{r})$ is the total electrostatic potential, and $\tilde{\rho}_0$ is chosen such that $\tilde{\rho}$ is normalised, i. e. $\int_{\tilde{r}} \tilde{\rho}(\tilde{r}) d\tilde{r} = N$. $\tilde{\rho}_0$ is determined most easily be finding an $\tilde{r}_0$ with $\tilde{\psi}(\tilde{r}_0) = 0$, since then $\tilde{\rho}_0 = \tilde{\rho}(\tilde{r}_0)$. Inserting this into Poisson's equation for $\tilde{\psi}$,

$$\Delta_{\tilde{r}} \tilde{\psi}(\tilde{r}) = 2 \left(2\pi\Xi\tilde{\rho}(\tilde{r}) - \tilde{\rho}_f(\tilde{r})\right) , \tag{7.13}$$

which is equivalent to Eqn. (7.10), one obtains the *Poisson–Boltzmann equation*

$$\Delta_{\tilde{r}} \tilde{\psi}(\tilde{r}) = 2 \left(2\pi\Xi\tilde{\rho}_0 e^{-\tilde{\psi}(\tilde{r})} - \tilde{\rho}_f(\tilde{r})\right) . \tag{7.14}$$

Since the key approximation in the Poisson–Boltzmann theory is the decoupling of the particles by using a product particle distribution, the Poisson–Boltzmann theory is only applicable to the case of small counterion–counterion coupling, i. e. small Ξ, see also the discussion in Ref. [66].

Application to a charged rod

Now we consider an infinitely long rod, which is neutralised by charged particles or counterions. We assume that the rod has a line charge density of $\tau < 0$ and radius R, and that the counterions have a charge of $q > 0$. The whole system is enclosed by a cylindrical simulation box of radius L (see Fig. 7.1). For this system, the fixed charge distribution is given by

$$\sigma(r) = \frac{\tau}{2\pi R} \delta(|r| - R) ,$$

which assumes that the rod is centred at $(0,0)$. Rescaling the coordinates by the Gouy–Chapman length of the rod surface

$$\mu = -\frac{R}{q\ell_B\tau} \tag{7.15}$$

leads to the following dimensionless coordinates and parameters

$$\tilde{R} = -\ell_B q\tau \quad \text{and} \quad \Xi = -q^3\ell_B^2\tau/R\,. \tag{7.16}$$

Note that the rescaled rod radius $\tilde{r}$ is *not* dependent on the rod radius itself.

Due to the radial symmetry, the Poisson–Boltzmann equation for this system is effectively one dimensional and reads for $\tilde{R} < \tilde{r} < \tilde{L}$ in the radial direction

$$\frac{d^2}{d\tilde{r}^2}\tilde{\psi}(\tilde{r}) + \frac{1}{\tilde{r}}\frac{d}{d\tilde{r}}\tilde{\psi}(\tilde{r}) = \kappa^2 e^{\tilde{\psi}(\tilde{r})}\,, \tag{7.17}$$

where $\kappa = \sqrt{4\pi\Xi\tilde{\rho}(\tilde{L})}$, which enforces $\tilde{\psi}(\tilde{L}) = 0$ as the normalisation. The rod charge enters only via the boundary conditions, which are obtained from Gauss' theorem as

$$\frac{d}{d\tilde{r}}\tilde{\psi}(\tilde{R}) = -2 \quad \text{and} \quad \frac{d}{d\tilde{r}}\tilde{\psi}(\tilde{L}) = 0\,. \tag{7.18}$$

The Poisson–Boltzmann equation can be solved exactly [32, 2], yielding

$$\tilde{\psi}(\tilde{r}) = -2\log\left(\frac{\tilde{r}}{\tilde{L}}\sqrt{1+C^{-2}}\cos\left(C\log\frac{\tilde{r}}{\tilde{R}_M}\right)\right)\,, \tag{7.19}$$

which can be verified easily by insertion. κ is related to C through the simple equation $\kappa^2\tilde{L}^2 = 2(1+C^2)$, but C and $\tilde{R}_M$ have to be determined from the boundary conditions (7.18), which leads to two coupled transcendental equations:

$$\tan\left(C\log\frac{\tilde{R}}{\tilde{R}_M}\right) = \frac{1-\tilde{R}}{C} \quad \text{and} \quad \tan\left(C\log\frac{\tilde{L}}{\tilde{R}_M}\right) = \frac{1}{C}\,, \tag{7.20}$$

which result in the equation

$$C\log\frac{\tilde{L}}{\tilde{R}} = \arctan\frac{1}{C} + \arctan\frac{\tilde{R}-1}{C} \tag{7.21}$$

for C. The fraction of counterions found in the region from radius $\tilde{R}$ to radius $\tilde{r}$ is given by

$$\phi(\tilde{r}) = \frac{1}{\tilde{\tau}}\int_{\tilde{R}}^{\tilde{r}} 2\pi\tilde{r}'\tilde{\rho}(\tilde{r}')d\tilde{r}' = 1 - \frac{1}{\tilde{R}} + \frac{C}{\tilde{R}}\tan\left(C\log\frac{\tilde{r}}{\tilde{R}_M}\right)\,, \tag{7.22}$$

where $\tilde{\tau} = \tau\mu$ is the rescaled line charge density. Note that $\phi(\tilde{R}_M) = 1 - 1/\tilde{R}$, which is an important quality of $\tilde{R}_M$, that will be described now.

Manning condensation

In the limit of infinite dilution, i. e. $\tilde{L} \to \infty$, the partition function for a single counterion is given up to some factors by

$$\int_{\tilde{R}}^{\infty} e^{-2\tilde{R}\log(\tilde{r})} 2\pi\tilde{r}d\tilde{r} = 2\pi \int_{1}^{\infty} x^{1-2\tilde{R}} dx\,, \tag{7.23}$$

which only exists for $\tilde{R} > 1$. Therefore the distribution function p_1 cannot be normalised for $\tilde{R} \leq 1$, and all particles diffuse away from the rod in the limit of infinite dilution. In contrast, rods with $\tilde{R} > 1$ are able to localise counterions. This led Oosawa [76] and Manning [62, 63, 64] to the idea that such rods *condense* a certain fraction of the particles, so that the counterions together with the rod form a larger rod of a smaller line charge density . With a fraction of condensed counterions equal to $1 - 1/\tilde{R}$, the charge parameter $\tilde{R}$ of the resulting rod is 1, and consequently the remaining counterions diffuses away.

This concept has become known as *Manning condensation*, the charge parameter $\tilde{R}$ is called the *Manning parameter* and $\tilde{R}_M$ the *Manning radius*. The Manning parameter $\tilde{R}$ is commonly denoted by ξ in the literature and determines the strength of the counterion–rod interaction, as one can see from its definition $\tilde{R} = q\ell_B\tau$. It can be proven strictly that in the Poisson–Boltzmann theory any fraction of the particles below $1 - 1/\xi$ will stay within a finite radius around the rod, while any fraction larger than $1 - 1/\xi$ spreads over the full space in the limit of infinite dilution [52].

A similar argument can be made for charged planes and charged spheres. In the first case, the counterions never diffuse away independently of the surface charge density, since the electrostatic potential of the plane, $\tilde{z}$, increases stronger than the logarithmic potential of the rod. In the case of a charged sphere the counterions cannot be bound at all, since the potential $-1/(2\pi\tilde{r})$ approaches a constant for $\tilde{r} \to \infty$. In this way the infinite rod case represents the transition between full or no counterion condensation, and the behaviour becomes dependent on the prefactors.

Using the concept of counterion condensation, the following simple argument states that attraction between two parallel like–charged rods can only occur if the combined charge parameter 2ξ of the two rods is larger than one [83]. From large distances, the two rods appear as one rod with twice the line charge density of a single rod. If the charge parameter of this double rod is smaller than one, the counterions will diffuse away in the limit of infinite dilution, and the two rods will only feel their bare repulsion. Although this provides some argument that attraction can occur only for $\xi > 1/2$, if at all, this argument cannot predict attraction directly.

Besides this simple argument of Manning, one can apply the Poisson–Boltzmann theory to the problem of attraction between like–charged macroions. However, also the Poisson–Boltzmann theory is not able to predict attraction [72, 95, 94, 85]. The Poisson–Boltzmann theory can give predictions for the particle density which are too low to compensate for the pure electrostatic repulsion of the rods, so that no attraction can occur at all. However, having sufficiently many particles close to the rods does not necessarily mean attraction in turn, and indeed the prediction of attraction is only possible, if counterion–counterion correlations are included.

The Poisson–Boltzmann theory is only valid for weak counterion–counterion coupling, in which case one does not expect attraction of the rods anyways. In contrast to this, the strong coupling theory is now presented, which is exact only in the opposite limit of infinite coupling parameter and is therefore better suited to predict attraction between like–charged objects.

7.2 Strong Coupling Theory

The strong coupling (SC–) theory of A. Moreira and R. Netz is obtained from a systematic expansion of the grand canonical free energy in inverse powers of Ξ [65]. Originally the SC theory was developed for the case of infinitely large charged planes, and tested extensively by Monte Carlo simulations [67], but recently the strong coupling theory was applied also to the case of two charged rods rsp. spheres, neutralised by their counterions [70, 69]. The derivation of the SC theory is fairly involved and will not be repeated here; for details on the SC theory, see Ref. [66]. Now only the main results will be given, namely the strong coupling free energy approximation and the predicted particle distribution.

The total potential energy of the system given in Eqn. (7.4) is split up into three parts, namely

$$\tilde{U} = \sum_{i<j} \frac{\Xi}{|\tilde{r}_i - \tilde{r}_j|} + \sum_i \tilde{u}(\tilde{r}_i) + \frac{\tilde{U}_0}{\pi\Xi}\,, \tag{7.24}$$

where the interaction of a particle with the fixed charge density is given by

$$\tilde{u}(\tilde{r}) = -\frac{1}{2\pi} \int\limits_{\tilde{r}'} \frac{\tilde{\rho}_f(\tilde{r})}{|\tilde{r} - \tilde{r}'|} d\tilde{r}'\,, \tag{7.25}$$

and the self interaction of the fixed charge density is

$$\tilde{U}_0 = \frac{1}{8\pi} \int\limits_{\tilde{r}} \int\limits_{\tilde{r}'} \frac{\tilde{\rho}_f(\tilde{r})\tilde{\rho}_f(\tilde{r}')}{|\tilde{r} - \tilde{r}'|} d\tilde{r}' d\tilde{r}\,. \tag{7.26}$$

The rescaled total charge of the system is given by

$$\tilde{Q} := \int\limits_{\tilde{r}} \tilde{\rho}_f(\tilde{r}) d\tilde{r} = -2\pi\Xi N\,, \tag{7.27}$$

where the second equality follows from the charge neutrality condition. Using these notations, the *strong coupling free energy* is given by

$$F_{SC} = \frac{F_1}{\Xi}\,, \tag{7.28}$$

where

$$F_1 = \left(\frac{\tilde{U}_0}{\pi} - \frac{\tilde{Q}}{2\pi} \log \int\limits_{\tilde{r}} e^{-\tilde{u}(\tilde{r})} + \frac{\tilde{Q}}{2\pi} \log \tilde{Q} - \frac{\tilde{Q}}{2\pi} \right) \tag{7.29}$$

is the leading order coefficient of a formal expansion of the free energy in terms of inverse powers of Ξ. Therefore the strong coupling energy should be exact in the limit

of infinite Ξ. But unlike F_1, all higher order coefficients (which are not shown here), dependent on Ξ; the expansion is not a power series in the strict sense. For the SC free energy to be exact in this limit it is sufficient if these coefficients are bounded for $\Xi \to \infty$. For the case of two charged planes this can be proven rigorously, but for other cases such as two rods or spheres this is unknown so far.

One should notice that F_1 only contains the macroion–macroion and macroion–counterion interactions, but not the counterion–counterion interactions, although the strong coupling theory is derived for the limit of infinite counterion coupling. In fact, due to the infinite counterion coupling, there is a huge correlation hole around each of the counterions, so that the concrete nature of the counterion–counterion interactions becomes unimportant. For example, at a charged surface the counterions will form a quasi crystal parallel to the charged plane. However, perpendicular to the plane, the counterion–counterion correlations only play a minor role, since the perpendicular distance of the counterions is small compared to their lateral distance. Therefore the particle density will not depend on the individual nature of the counterion–counterion correlations in the strong coupling limit. A similar argument can be formulated for the case of two charged rods or spheres.

To renormalise the potential energy contributions, one chooses a convenient fixed reference position $\tilde{r}_0$ and replaces $\tilde{U}_0$ by

$$\tilde{U}_0^s = \frac{1}{8\pi} \int_{\tilde{r}} \int_{\tilde{r}'} \frac{\tilde{\rho}_f(\tilde{r})\tilde{\rho}_f(\tilde{r}')}{|\tilde{r}-\tilde{r}'|} d\tilde{r}' d\tilde{r} - \frac{\tilde{Q}}{4\pi} \int_{\tilde{r}} \frac{\tilde{\rho}_f(\tilde{r})}{|\tilde{r}-\tilde{r}_0|} d\tilde{r} \tag{7.30}$$

and $\tilde{u}$ by

$$\tilde{u}^s(\tilde{r}) = -\frac{1}{2\pi} \int_{\tilde{r}'} \tilde{\rho}_f(\tilde{r}) \left(\frac{1}{|\tilde{r}-\tilde{r}'|} - \frac{1}{|\tilde{r}_0-\tilde{r}'|} \right) d\tilde{r}' \,. \tag{7.31}$$

As one can see clearly from Eqn. (7.31), the additional contribution is the interaction of the fixed charge density with all particles located at $\tilde{r}_0$. These modifications cancel in the total electrostatic energy (7.24) since

$$\begin{aligned} \frac{1}{\pi\Xi} \left(\tilde{U}_0^s - \tilde{U}_0 \right) &= -\frac{\tilde{Q}}{4\pi^2\Xi} \int_{\tilde{r}} \frac{\tilde{\rho}_f(\tilde{r})}{|\tilde{r}-\tilde{r}_0|} d\tilde{r} \\ &= -\frac{N}{2\pi} \int_{\tilde{r}} \frac{\tilde{\rho}_f(\tilde{r})}{|\tilde{r}-\tilde{r}_0|} d\tilde{r} = -\sum_i \left(\tilde{u}^s(\tilde{r}_i) - \tilde{u}(\tilde{r}_i) \right) \,. \end{aligned} \tag{7.32}$$

Therefore $\tilde{U}_0$ and $\tilde{u}$ can be replaced by $\tilde{U}_0^s$ and $\tilde{u}^s$ in Eqn. (7.29) without changing the value.

Similarly to the free energy, one can obtain the particle distribution in the limit of infinite Ξ, the *strong coupling particle distribution*, as

$$\tilde{\rho}(\tilde{r}) = \frac{e^{-\tilde{u}(\tilde{r})}}{\int_{\tilde{r}} e^{-\tilde{u}(\tilde{r}')} d\tilde{r}'} \tag{7.33}$$

for all $\tilde{r}$ which are accessible to the particles. The particle distribution depends only on the macroion–counterion interactions, but not on the counterion–counterion interactions, which is in correspondence with the free energy, see the discussion above.

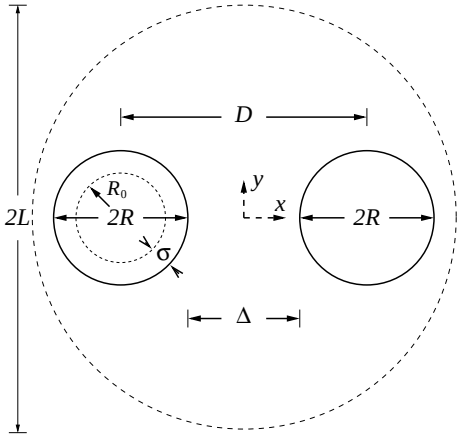


Figure 7.2: Schematic view of the two rod system from along the z–axis. R is the radius of the rod. In the simulations it consists of the offset radius R_0 plus the soft repulsion radius σ. The centre–centre distance of the rods is D, the cylindrical simulation cell has radius L.

Application to the two–rod problem

For the two–rod problem the fixed charge density of two rods of radius R and line charge density $\tau < 0$ is given by

$$\sigma(r) = \tau(\delta(\rho_1 - R) + \delta(\rho_2 - R))\,,$$

where the radial distances to the rod centres are

$$\rho_{1,2} = \sqrt{(x \pm D/2)^2 + y^2}\,.$$

$D = 2R + \Delta$ is the centre–centre distance of the two rods (see Fig.7.2), Δ their surface to surface distance. The particles have charge q and live in a region outside the two rods, but inside a cylindrical compartment of radius L. The strong coupling theory was applied to the two rod problem first by Naji and Netz [70].

The Gouy–Chapman length of this system is the same as for the single rod, $\mu = -R/(q\ell_B\tau)$, and the rescaled coordinates are the same as in Sec. 7.1. Choosing $\tilde{r}_0 = 0$, we obtain the self interaction of the fixed charged density per rescaled length of the rod $\tilde{H}$, as

$$\frac{\tilde{U}_0^s}{\tilde{H}} = -2\pi\tilde{R}^2 \log\tilde{D} + 2\frac{\tilde{Q}}{\tilde{H}} \log\left(\frac{\tilde{D}}{2}\right) = 6\pi\tilde{R}^2 \log\tilde{D} - 8\pi\tilde{R}^2 \log 2$$

up to the irrelevant self interactions of both rods. Note that due to the choice of $\tilde{r}_0$, $\tilde{U}_0^s$ corresponds to the electrostatic energy of the macroions with themselves and all the counterions are located at the centre of the system between the two rods (compare Eqn. (7.30)). For this configuration, the electrostatic energy $\tilde{U}_o^s$ is attractive. The particle–rod interactions are given by

$$\tilde{u}^s(\tilde{r}) = -2\tilde{R}\left(\log\tilde{\rho}_1 + \log\tilde{\rho}_2 - \log(\tilde{\Delta}^2/4)\right)\,.$$

The leading order coefficient of the free energy is up to an irrelevant additive constant given by

$$\frac{F_1}{\tilde{H}} = -2\tilde{R}^2 \log\tilde{D} - 2\tilde{R}\log I\,, \tag{7.34}$$

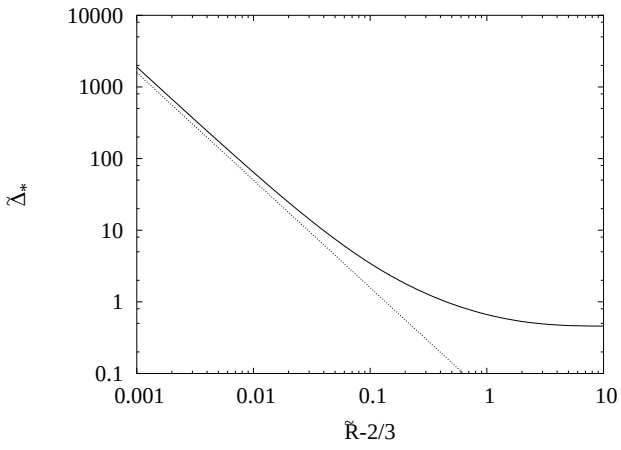


Figure 7.3: SC prediction for the threshold of attraction in the case of infinite $\tilde{L}$. The graph shows the equilibrium surface to surface distance $\tilde{\Delta}_*$ as a function of the rescaled rod radius $\tilde{R} - 2/3$. The dotted line marks the asymptotic curve $(\tilde{R} - 2/3)^{-3/2}$.

where the first term represents the bare rod–rod repulsion, while

$$I = \int_{\tilde{r}} e^{-2\tilde{R}(\log\tilde{\rho}_1 + \log\tilde{\rho}_2)} d\tilde{x}d\tilde{y} = \int_{\tilde{r}} (\tilde{\rho}_1\tilde{\rho}_2)^{-2\tilde{R}}\, d\tilde{x}d\tilde{y} \tag{7.35}$$

represents the counterion–rod correlations. I is integrated over the rather complex area accessible to the particles and therefore can be calculated numerically only. Nevertheless some general arguments on its scaling are possible. For a charge parameter of $\xi = \tilde{R} < 1/2$ the integral is divergent in the limit of infinite dilution, and no attraction between the rods occurs. This is in agreement with the simple argument using the concept of Manning condensation presented in Sec 7.1.

For large $\tilde{D}$ we rescale the coordinates in I by $\tilde{D}$, obtaining

$$I = \tilde{D}^{2-4\tilde{R}} \int_{\tilde{r}} \left(\left(\left(\tilde{x}_d + \frac{1}{2} \right)^2 + \tilde{y}_d^2 \right) \left(\left(\tilde{x}_d - \frac{1}{2} \right)^2 + \tilde{y}_d^2 \right) \right)^{-\tilde{R}} d\tilde{x}_d d\tilde{y}_d \,,$$

where the integral is bounded in $\tilde{D}$. Therefore

$$\frac{F_1}{\tilde{H}} = -2\tilde{R}(2 - 3\tilde{R}) \log \tilde{D} + \mathcal{O}_{\tilde{D}}(1) \,. \tag{7.36}$$

From this immediately follows that attraction occurs for all $\tilde{R} > 2/3$. A more involved calculation shows that for $\tilde{R}$ close to $2/3$, the equilibrium surface to surface distance is of order $\tilde{\Delta}_* = \mathcal{O}(\tilde{R} - 2/3)^{3/2}$ [70].

While Eqn. (7.35) shows that attraction can only occur in the presence of condensed counterions, i. e. $\tilde{R} = \xi > 1/2$, Eqn. (7.36) shows that counterion condensation is not sufficient for attraction to occur. For all $\tilde{R}$, $\mathcal{F}$ has at most one minimum with respect to $\tilde{\Delta}$ (so for $\tilde{R} > 2/3$ precisely one). Therefore for all $\tilde{R} > 2/3$ attraction between the rods occurs, and the equilibrium distance between them is finite. The threshold of attraction has to be determined by numerical integration and is show in Fig. 7.3.

The SC counterion distribution is given by

$$\tilde{\rho}(\tilde{r}) = \frac{(\tilde{\rho}_1\tilde{\rho}_2)^{-2\tilde{R}}}{\int_{\tilde{r}'} (\tilde{\rho}'_1\tilde{\rho}'_2)^{-2\tilde{R}}\, d\tilde{r}'} \tag{7.37}$$

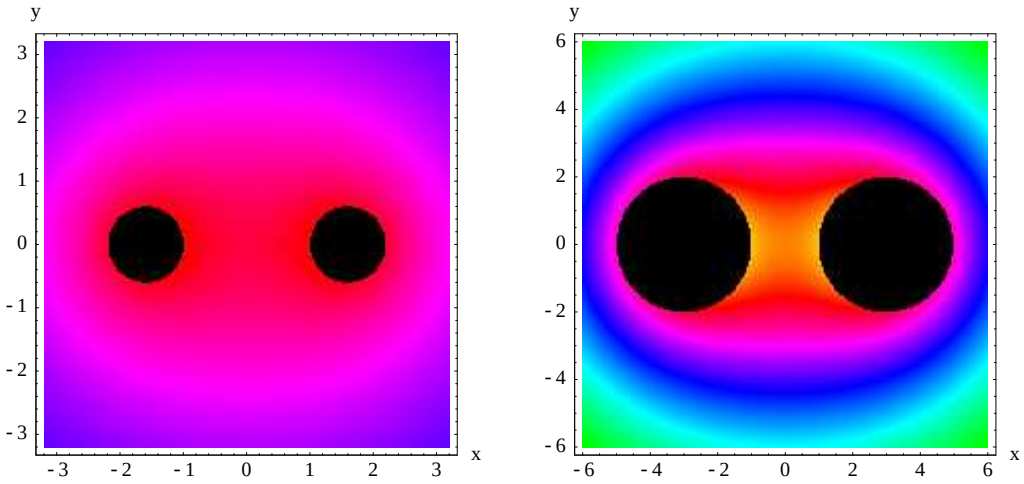


Figure 7.4: SC prediction for the counterion density for $\tilde{R} = 0.6$, $\tilde{\Delta} = 2$ (left) and $\tilde{R} = \tilde{\Delta} = 2$ (right). The black circles mark the rods. The density increases from green to red, i. e. the particles accumulate on the rod surfaces in the gap between the rods. The colour scale is the same for both graphs, i. e. for $\tilde{R} = 2$ the particles are much closer to the rods than for $\tilde{R} = 0.6$; for $\tilde{R} = 2$ about 95% of the particles are located in the red region, while for $\tilde{R} = 0.6$ this fraction is only about 20%.

for all positions $\tilde{r}$ accessible to the counterions. Two resulting counterion densities are shown in Fig. 7.4. As one can see, the counterions accumulate primarily on the rod surfaces pointing towards the opposite rod. For sufficiently large $\tilde{R}$, even in the limit of infinite dilution almost all counterions can be found close to the rod surfaces and in the gap between the rods. Therefore although the SC theory is obtained in the limit of infinite counterion coupling, the counterions are packed quite densely onto the rod surfaces. This can be understood from the fact that counterion correlations perpendicular to the macroion–surfaces are unimportant in the strong coupling limit, provided the higher coefficients $F_2, \ldots$ of the strong coupling expansion do not contribute in the limit of infinite Ξ, see the discussion above. Therefore in this direction the macroion–counterion correlations dominate, which attract the counterions to the rod surfaces.

It is not yet known how the higher order coefficients in the expansion of the free energy, i. e. $F_2, F_3, \ldots$, are bounded or not. However, these higher order terms are only of importance if counterion–counterion interactions are non-negligible. Rouzina and Bloomfield [84] suggested as a simple criterion for the importance of counterion–counterion interactions the parameter

$$\gamma_{RB} = -q/(\tau\Delta) < 1\,. \tag{7.38}$$

In other words, counterion–counterion interactions become important if the average counterion–counterion distance along the rod axis, $-q/(2\tau)$, is of the same order as the surface to surface distance of the rods Δ or even smaller. This is reasonable, since in the strong coupling limit, the counterions are correlated strongly along the rod axes. For large Rouzina–Bloomfield parameter, the lateral distance of the counterions in the gap between the rods is therefore much larger than their perpendicular

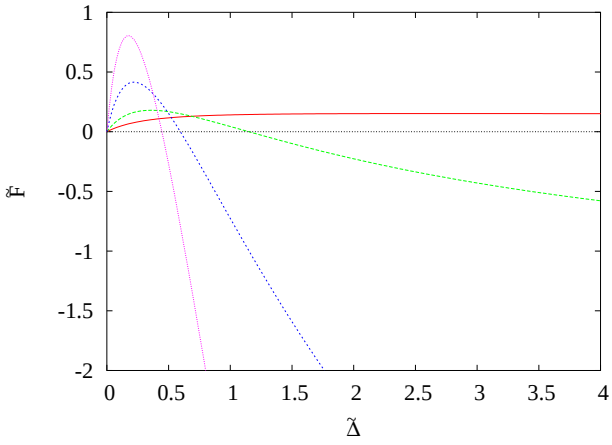


Figure 7.5: Example rescaled force $\tilde{F}$ curves as a function of $\tilde{\Delta}$ for $\tilde{R} = 0.5$ (solid red), $\tilde{R} = 1$ (dashed green), $\tilde{R} = 2$ (dashed blue) and $\tilde{R} = 4$ (dotted magenta). For $\tilde{R} = 0.5$ no second zero exists. These curves are sampled in the numerical simulations and their positive zero determined.

distance, so that the counterion–counterion interactions are indeed unimportant for the perpendicular particle distribution, see also the discussion of F_1 above.

For real DNA with trivalent counterions, such as spermidine, the average counterion–counterion distance along the rod axis is $q/(2\tau) \approx 2.5$Å, so that the strong coupling theory is only valid if two DNA strands are much closer than 5Å. But since water at room temperature has a Bjerrum length of $\ell_B \approx 7$Å and the DNA radius is $R \approx 10$Å, DNA has a Gouy–Chapman length of $\mu \approx 0.8$Å and a rescaled rod radius of $\tilde{R} \approx 12.6$. For this high rescaled rod radius, the predicted equilibrium surface to surface distance is below μ, i. e. well in the range of a high Rouzina–Bloomfield–parameter, so that the strong–coupling theory is valid, and indeed can be used to predict attraction between like–charged rods.

Interestingly we have

$$\Xi = -q^3 \ell_B^2 \tau / R = \gamma_{RB} \tilde{R} \tilde{\Delta}\,, \tag{7.39}$$

so that for fixed $\tilde{R}$ and $\tilde{D}$ γ_{RB} determines Ξ. We will see later that γ_{RB} can be used as a parameter to describe the ability of the SC theory to predict the equilibrium distance of two rods at finite Ξ.

7.3 Comparison with numerical results

This section presents results of computer simulations for the equilibrium surface to surface distance as well as for the particle distribution which are compared to the predictions of the strong coupling theory.

Simulation method

To obtain the equilibrium rescaled surface to surface distance, i. e. the distance with zero average force between the rods, a series of simulations with either fixed $\tilde{R}$ or fixed $\tilde{\Delta}$ was done. While fixing $\tilde{R}$ and varying $\tilde{\Delta}$ allows to sample the equilibrium surface to surface distance for larger $\tilde{R}$, since the curve is more or less constant there, for $\tilde{R} \to 2/3$ the curve is to steep (compare Fig. 7.3). Here the sampling with fixed $\tilde{\Delta}$ and varying $\tilde{R}$ is more robust.

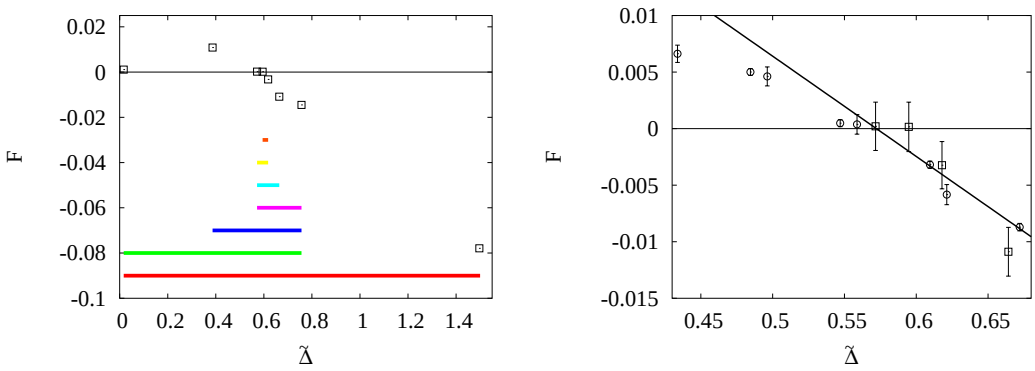


Figure 7.6: Example of the bisection and interpolation for parameters $\Delta = 0.5$, $q = 3$, $\tau = 0.1$ and $\tilde{R} = 3$. The data points give the measured force F as function of $\tilde{\Delta}$. The left graph shows the data points acquired during the bisection, the coloured bars denote the generated intervals starting at the bottom. The right graph shows the points used during the interpolation, including the results of two additional runs (circles).

In the previous section it was discussed that the energy curve has at most one minimum. The force curve has at most two zeros, one at zero, which is of little interest here, and eventually a second one at the equilibrium rescaled surface to surface distance (rsp. the equilibrium rescaled radius if the rescaled surface to surface distance was kept fixed). Since the average force between the rods is continuous with respect to $\tilde{R}$ rsp. $\tilde{\Delta}$, there are many methods to determine its zero, the most prominent ones are the bisection method and the regula falsi.

In general, the regula falsi is faster than the bisection method. However, between the two zeros there is a maximum, which comparatively close to the two zeros (see Fig. 7.5). This situation is inconvenient for the regula falsi, since it tends to overshoot when close to the maximum. This problem is enhanced by the fact that the data points are not exact. The overshooting can be reduced by damping, at the expense of slower convergence. Moreover, there is no simple way to keep the regula falsi from converging to $\tilde{R} = 0$, which we are not interested in.

Therefore in this work, the bisection algorithm was used. It is simple, robust, and cannot converge to $\tilde{R} = 0$ by construction, which more than compensates for its weak convergence. Whenever the sign of the force at a bisection point is positive, the bisection algorithm assumes that the zero cannot below the bisection point, which inhibits the convergence to $\tilde{R} = 0$. For the bisection algorithm only the sign of the force is needed, for which the probability of a wrong value is highest close the correct zero, assuming a practically constant force error width.

Therefore the result interval of the bisection algorithm should be close to the $\tilde{R}_0$. In the vicinity of $\tilde{R}_0$, the force is reasonably linear, which allows to use a generalised form of the regula falsi to improve on the zero. In this work, in an interval of width 0.25 around the resulting zero of the bisection algorithm, additional data points were generated. To all data points in this interval, a straight line was fitted via weighted linear regression, and its zero used as improved guess. The errorbars of the resulting zero guesses were calculated from the error estimates for the linear regression via error

propagation. See Fig. 7.6 for an example of the combined bisection and regula falsi.

The data points, i. e. the average forces between the two rods, were obtained from NVT MD simulations. These simulations as well as the bisection algorithm were performed by a single ESPResSo script. The interpolation and additional simulations were also performed by ESPResSo/Tcl scripts, which again demonstrates the power of the scripting concept. Simulations and analysis were run on single AMD MP 2000+ processors, accumulating a computation time of around 8000 hours. The compressed configuration data amounts to about 6 Gigabyte.

For the simulations the standard Verlet integrator combined with a Langevin thermostat were used, see Sec. 1.4. The electrostatic interactions were calculated using the MMM1D formulas tuned to a maximal pairwise force error of 10^{-4}, see Sec. 6.4. The excluded–volume interactions between the rods and the point–like counterions were modelled using the purely repulsive Lennard–Jones potential with a fixed soft repulsion radius of $\sigma = 1$ and a variable offset R_0, see Sec. 1.5. Assuming that the counterions cannot penetrate the Lennard–Jones potential by more than σ, the effective rod radius is $R = R_0 + \sigma$, as shown in Fig. 7.2.

Since the first few data points generated in the bisection algorithm are probably far away from the force zero, they will not be used for the linear regression, and even with a high force error the force sign will probably be correct. Therefore for the first centre point of the bisection only 100 configurations were generated, for the next centre points 600, 1100, 1600 and 2000 for the fifth one and all following. The additional simulation runs in the vicinity of the bisection guess for the zero all generated 20000 configurations.

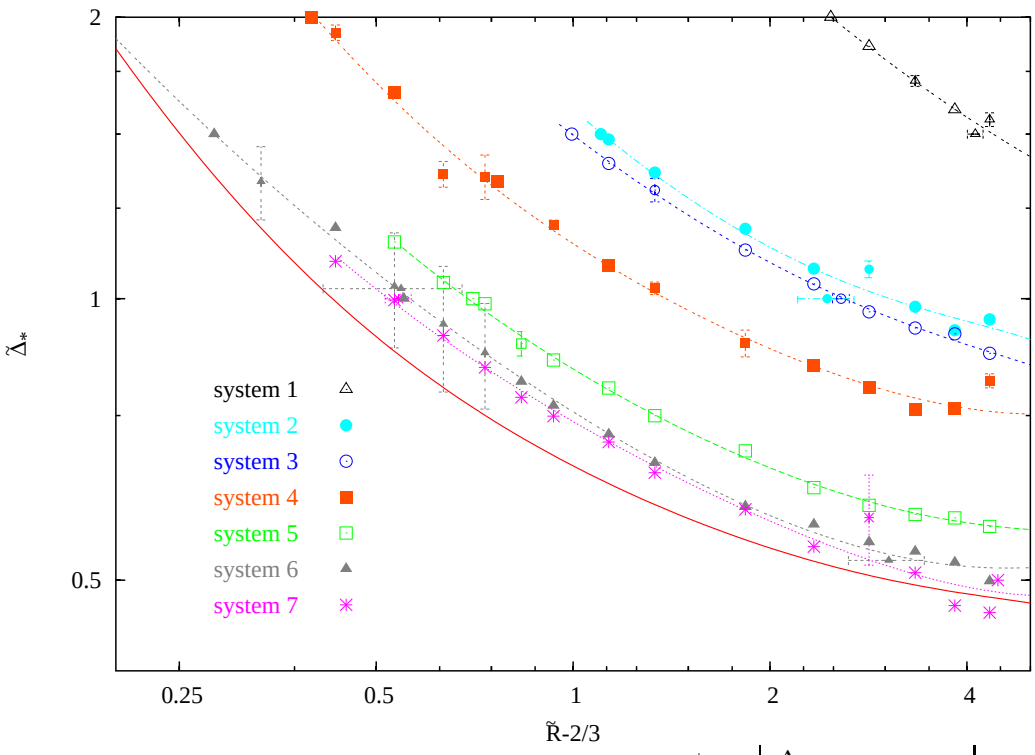
The number of counterions in the simulations was 24. Since the simulations were performed with small line charge densities τ to achieve a reasonably high $\gamma_{RB} = q/(\tau\Delta)$, one doesn't expect strong ion–ion correlations along the axis parallel to the rods and therefore a large number of counterions is not necessary. This was confirmed by a test simulation using 48 counterions which showed no abnormal deviations from the 24 counterion simulations. On the other hand, increasing the number of counterions increases the computation time dramatically, since MMM1D is a N^2–method and cannot be parallelised efficiently.

For a single run the surface to surface distance Δ, the counterion charge q and line charge density τ were fixed, i. e. the Rouzina–Bloomfield parameter γ_{RB} was constant. Then for the bisection the rod radius R and Bjerrum length ℓ_B were calculated from $\tilde{R}$ and $\tilde{\Delta}$ using the Eqns. 7.16, which give

$$\ell_B = \tilde{R}/(-q\tau) \quad \text{and} \quad R = \Delta\frac{\tilde{R}}{\tilde{\Delta}} \tag{7.40}$$

$\tilde{\Delta}$ is used to determine R instead of Δ to ensure that Δ does not become too small, even if $\tilde{\Delta}$ is small. If Δ is much smaller than the soft repulsion radius σ, the soft nature of the Lennard–Jones potential would come into play. More importantly, both Ξ and γ_{RB} are not constant throughout the simulation if for example the rod radius is changed. In this way one could not expect any agreement between theory and simulations. In fact, we will see in a moment that indeed simulations at constant Ξ show strong deviations from the theoretical prediction.

The radius of the cylindrical simulation box was chosen to be $L = 4D$. The SC



system	Δ	τ	q	γ_{RB}
1	1	1	3	3
2	1	0.33	3	9.1
3	1	0.1	1	10
4	2	0.1	3	15
5	1	0.1	3	30
6	2.5	0.1	10	40
7	1	0.1	5	50

Figure 7.7: Simulation results for the equilibrium surface to surface distance $\tilde{\Delta}_*$ as a function of the rescaled rod radius $\tilde{R} - 2/3$ at various parameters, see the table to the right. The solid line gives the SC prediction. Error bars are only shown if they are larger than the symbol.

results for this choice differ from the results at infinite simulation box radius by at most 5%, so that this choice is close to an infinite simulation box.

The soft potential implies that $R \geq \sigma = 1$, so that the additional constraint

$$\tilde{R}\Delta/\tilde{\Delta} = R \geq \sigma \quad \text{rsp.} \quad \tilde{\Delta} \leq \Delta\tilde{R}/\sigma \tag{7.41}$$

arises. This leads to the requirement that Δ is not too small, although this decreases the Rouzina–Bloomfield parameter and has to be compensated by a higher counterion charge or smaller line charge density.

Onset of attraction results

Fig. 7.7 shows the simulation results for the equilibrium surface to surface distance for various parameters, each with fixed γ_{RB}. One can see that indeed γ_{RB} determines the level of agreement between the simulation curve and the theoretical prediction. Low values of γ_{RB} correspond to curves far away from the SC prediction, while the curves with $\gamma_{RB} > 40$ are already close the strong coupling curve.

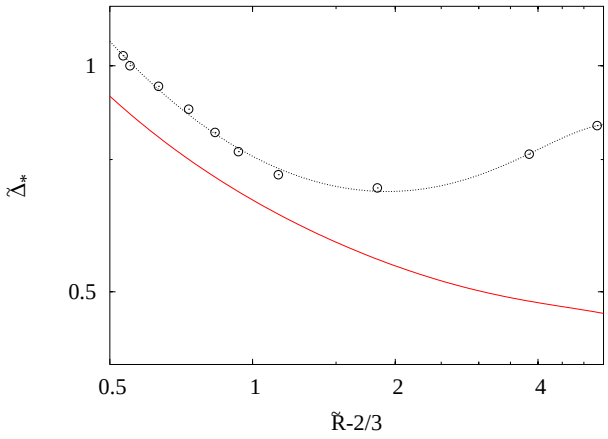


Figure 7.8: Data points from Fig. 7.7 with $|\Xi - 50| \leq 5$. The solid line gives again the SC prediction.

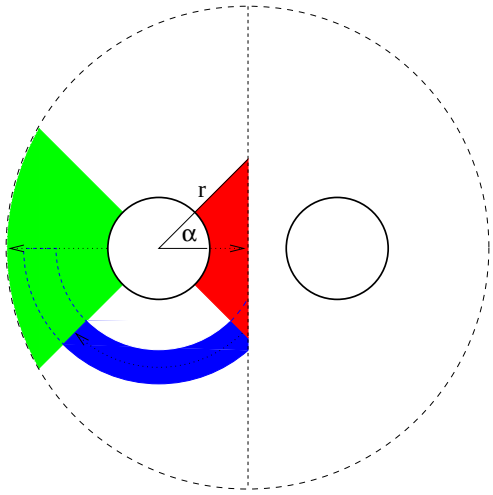


Figure 7.9: Schematic view of the paths along which the densities are measured. The coordinate system is cylindrical around the left rod, with angle $\alpha = 0$ pointing towards the right rod. Radial densities are measured in two sectors ranging from $-\pi/4$ to $\pi/4$ (red area) and from $3\pi/4$ to $5\pi/4$ (blue area), i. e. pointing towards the other rod and away from it. Azimuthal densities are measured in the radius ranges $R + \Delta/4$ to $R + 3\Delta/4$, $R + \Delta \pm \Delta/4$ and $R + 4\Delta \pm \Delta/4$. The dotted arrows denote the direction along which the densities are measured.

Similar γ_{RB}, for example $\gamma_{RB} = 9$ and $\gamma_{RB} = 10$ or $\gamma_{RB} = 40$ and $\gamma_{RB} = 50$, produce similar curves. For this it is unimportant how the value of γ_{RB} is obtained, whether by changing the line charge density, counterion charge or surface to surface separation. The latter pair of curves do not differ much, although the counterion charge differs by a factor of 2, as well as the surface to surface distance.

The computer simulations therefore strongly support γ_{RB} as a measure for the degree of validity of the SC theory at finite Ξ rsp. for the importance of counterion correlations, as proposed by Rouzina and Bloomfield. Interestingly, Ξ is *not* an equally good measure of the fit, although Ξ is the expansion parameter. Since $\Xi = \gamma_{RB}\tilde{\Delta}\tilde{R}$, constant Ξ results in $\gamma_{RB} \sim 1/(\tilde{R}\tilde{\Delta})$, i. e. lower γ_{RB} for higher $\tilde{R}$. Fig. 7.8 shows selected data points from Fig. 7.7, for which $|\Xi - 50| \leq 5$. The resulting $\tilde{\Delta}_*$ grows with $\tilde{R}$, which is reasonable, since γ_{RB} decreases. Therefore Ξ does not determine the size of the higher order corrections alone, which shows that the prefactors of Ξ in the expansion cannot be constant, i. e. that the expansion is not a strict power series.

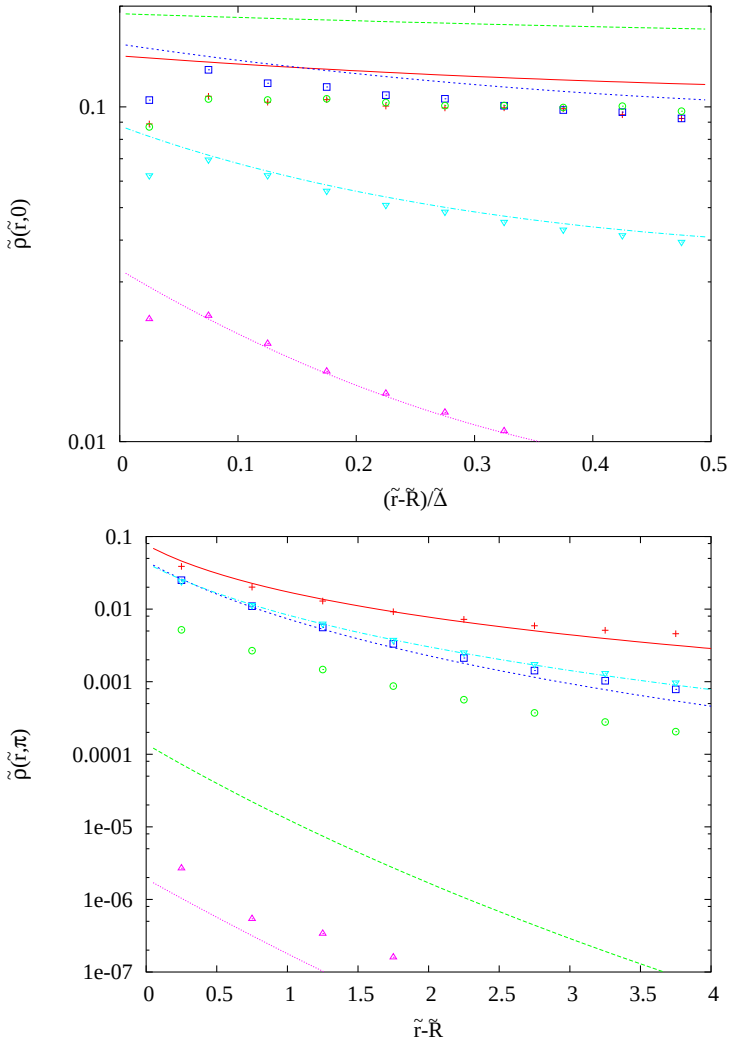


Figure 7.10: Rescaled radial ion distributions $\tilde{\rho}(\tilde{r}, \alpha)$ in directions $\alpha = 0$ (top, red area in Fig. 7.9) and $\alpha = \pi$ (bottom, gree area) as a function of the distance to the rod surface $\tilde{r} - \tilde{R}$. For $\alpha = 0$ the distance is given in multiples of the surface to surface distance $\tilde{\Delta}$. The simulation data represented by the symbols is taken from the run with $\Delta = 1$, $\tau = 0.1$ and $q = 5$, the lines give the SC predictions. The parameters are: $\tilde{R} = 7$, $\tilde{\Delta} = 5$ (magenta triangle ups), $\tilde{R} = 0.5$, $\tilde{\Delta} = 0.5$ (red crosses), $\tilde{R} = 4$, $\tilde{\Delta} = 0.5$ (green circles), $\tilde{R} = 1$, $\tilde{\Delta} = 1$ (blue squares), $\tilde{R} = 0.85$, $\tilde{\Delta} = 2$ (cyan triangle downs).

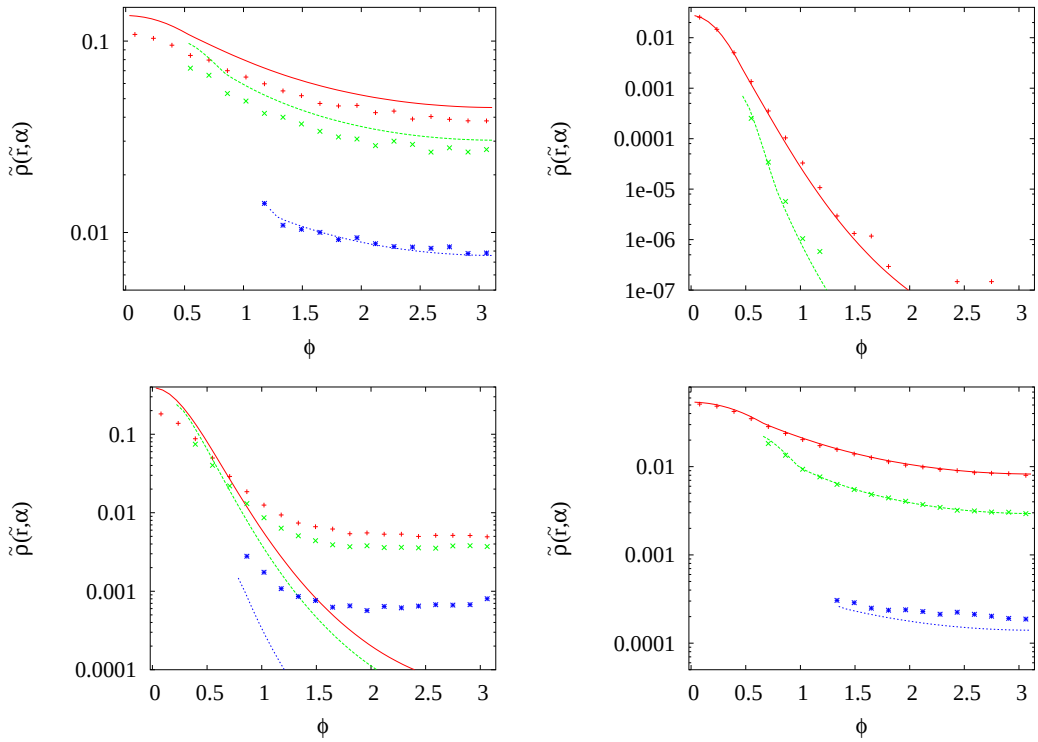


Figure 7.11: Azimuthal particle distribution $\tilde{\rho}(\tilde{r}, \alpha)$ as a function of the angle to the rod–rod axis α for $\tilde{R} = 0.5$, $\tilde{\Delta} = 0.5$ (top left), $\tilde{R} = 7$, $\tilde{\Delta} = 5$ (top right), $\tilde{R} = 4$, $\tilde{R} = 0.5$ (bottom left) and $\tilde{R} = 0.85$, $\tilde{\Delta} = 2$ (bottom right). The densities are given for radii $\tilde{r} = \tilde{R} + \tilde{\Delta}/2 \pm \Delta/4$ (red plusses), $\tilde{r} = \tilde{R} + \tilde{\Delta} \pm \Delta/4$ (green crosses) and $\tilde{r} = \tilde{R} + 4\tilde{\Delta} \pm \Delta/4$ (blue stars). Again the symbols denote results from computer simulations, the lines show the corresponding SC predictions.

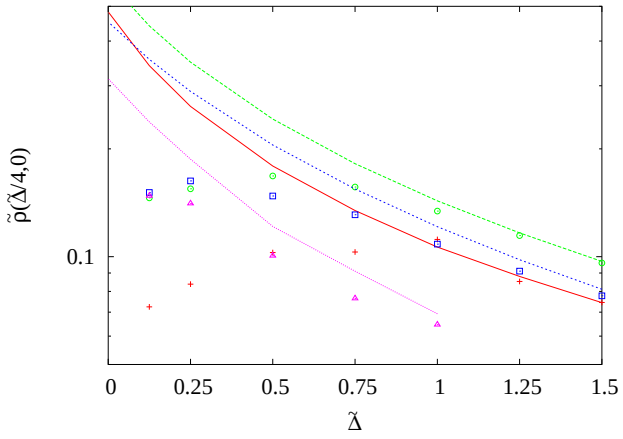


Figure 7.12: Radial density $\tilde{\rho}(\Delta/4, 0)$ at a distance of $\Delta/4$ from the centre towards one of the rods (i. e. $(\tilde{r} - \tilde{R})/\tilde{\Delta} = 0.25$ in Fig. 7.10), as a function of $\tilde{\Delta}$. Red crosses denote simulation data for $\tilde{R} = 4$, green circles for $\tilde{R} = 2$, blue squares for $\tilde{R} = 1$ and magenta triangles for $\tilde{R} = 0.5$. The lines give the SC predictions as usual.

Particle density results

The strong coupling particle distribution (see Fig. 7.4) shows an accumulation of the particles around the rods and especially in the gap between the two rods. Since one is interested in how well this accumulation is reproduced in the simulation, the density results of the simulations were compared to the strong coupling theory in three sections, as depicted in Fig. 7.9. The directions of the radial particle densities were chosen to cover both the region of highest particle density between the rods and the region of lowest particle density on the opposite side. The azimuthal densities compare the particle distribution both in the close vicinity of the rods as well as at a moderately high density.

Fig. 7.10 and 7.11 compare the strong coupling particle density to the densities obtained from the simulation with $\Delta = 2.5$, $q = 10$ and $\tau = 0.1$, i. e. $\gamma_{RB} = 40$. Although the density in the systems varies by several orders of magnitude, the results fit well to the SC predictions, with the exception of the particle densities for $\tilde{R} = 4$, $\tilde{\Delta} = 0.5$ and, to less extent, also $\tilde{R} = 0.5$, $\tilde{\Delta} = 0.5$, which will be discussed in a moment. The decreased counterion density close to the rod surface (the very left data points in Fig. 7.10) are an artifact of the soft repulsion potential between the rods and the particles.

Fig. 7.12 shows the particle density for a fixed point in the gap between the rods for various values of $\tilde{R}$ and $\tilde{\Delta}$. For $\tilde{\Delta} < 1$ the behaviour of the particle densities differs qualitatively between the SC prediction and the simulation results. While in the strong coupling theory the particle density monotonously grows with decreasing $\tilde{\Delta}$, the particle distribution obtained from the simulations have a maximum and drop again for small $\tilde{\Delta}$. This deviation cannot be attributed to the soft repulsion potential, since the coordinates are chosen such that the surface to surface distance in the simulations is constant, while the rod radius grows, see Eqn. 7.40.

To understand the difference at small $\tilde{\Delta}$, one has to remember that the strong coupling theory ignores counterion–counterion correlations in the directions perpendicular to the rod surfaces, as they are unimportant in the strong coupling limit. However, for finite coupling parameter they do play a role, preventing large local counterion densities. Therefore at small gap sizes $\tilde{\Delta}$ the density in the gap cannot grow as large as predicted by the strong coupling theory, while in the rest of the

system the density has to be higher to compensate for this.

These deviations in the particle densities become strong only below $\tilde{\Delta} = 3/4$, which is about the order of the predicted equilibrium surface to surface distance, so that the equilibrium system is described reasonably well by the strong coupling theory, although Ξ is only 40. For the case of two planes a similar agreement can only be achieved for Ξ larger than 10^4 (see Ref. [66]).

7.4 Zero Temperature

The limit of zero temperature corresponds to an infinite Bjerrum length and therefore infinite Ξ as well as infinite $\tilde{R}$ and $\tilde{\Delta} = \tilde{R}\Delta/R$. In this limit the Strong Coupling Theory predicts attraction for all parameter values at zero temperature, and the equilibrium surface to surface distance is 0. This is consistent with the simulations presented in the next section and can be easily understood from Earnshaw's theorem[1]. To further test the SC predictions, we again look at the counterion distribution function. The SC theory predicts that all particles will line up between the rods at zero temperature, since the exponent of the particle distribution in Eqn. (7.37) tends to infinity, resulting in a peak where both surfaces coincide.

Computer simulations, such as the ones presented later, show a more complex behaviour. Depending on the parameters, the particles can line up between the rods (see Fig. 7.20) or populate more space on the rod surfaces (see Figs. 7.23 and 7.24). Between these two extremes an interesting intermediate state exists where the particles are not lined up between the rods, but still do not leave the rod–rod plane (see Figs. 7.21 and 7.22). This planar state will be investigated in detail in the following. In the formulation of this model, the SC Theory prediction is that at zero temperature for all parameter values, $\alpha = 1$ is optimal.

The discrepancy between simulation results and the SC prediction can be attributed again to the neglected higher order terms of the particle density expansion. Obviously, the Rouzina–Bloomfield parameter $\gamma_{RB} = -q/(\tau\Delta)$ cannot be used to quantify the importance of these terms as it is infinite. In this section, a simple analytic approach is used to investigate the planar state and a replacement for the Rouzina–Bloomfield parameter for small Δ will be derived.

7.5 Analytical calculation

To study the planar state, which contains the lined–up state of the SC theory as a special case, we use a simplified model as shown in Fig. 7.13. We consider systems where all counterions are located in three rods, one in the middle between the two rods with a line charge density $2\alpha\tau$ and two rods of line charge density $(1-\alpha)\tau$ at the opposite sides of the rods. In the following the rods of line charge τ will be called *macroion–rods*, while the rods representing the particles will be called *particle–rods*.

The interactions between two different particle–rods are highly complex, since for arbitrary α the particles will not be equally spaced in the ground state. The MMM1D

[1] Earnshaw's theorem states that a collection of point charges cannot be kept in equilibrium by pure electrostatic interactions [26]. Therefore the rods have to be located surface to surface, so that the excluded volume interaction counters the electrostatic interactions.

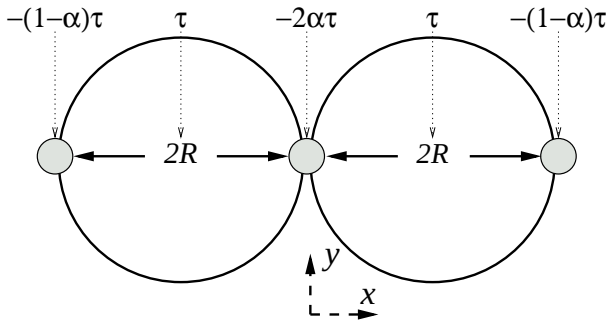


Figure 7.13: Schematic view of the simplified model as used in Sec. 7.5. The grey circles show the particle lines, which are treated as rods of the same line charge density, which is given below each rod. Unlike shown, the counterions have zero size and therefore the surfaces of the rods touch.

far formula (6.13) shows that the interaction between two particles is essentially rod–like plus a sum of Bessel terms, which drop faster than exponentially with increasing distance. Therefore it is reasonable to approximate the interactions of two different particle–rods by the interaction of homogeneously charged rods of equal charge.

This approximation is not valid for the interaction of a particle–rod with itself, i. e. the self energy of the particle–rods. Assuming that the charges are equally spaced, the self interaction of the particle–rod can be easily calculated using the MMM1D near formula (6.16), which will be used as the self energy of the particle–rods in this model, even though they may not be equally spaced in the correct ground state.

In the following it will be determined under which circumstances these configurations are stable and which value of α is optimal, i. e. has minimal energy. The stability of the configuration requires that the net force between the macroion–rods is attractive, so that they do not move apart, and that a particle in both the outer and the inner particle–rods is in a local energy minimum, so that none of the particles is able to move. However, this does not assure that the configuration with the optimal α is a global minimum of the original two rod problem. The computer simulations presented in the following section show similar approximate ground states, so that energies calculated using the simplified model seem to be at least close to the real ground state.

Force between the rods

In the planar model, the total force between the two macroion–rods is

$$2\tau^2/(2R) + \frac{4(1-\alpha)\tau^2}{R} - \frac{4(1-\alpha)\tau^2}{3R} - \frac{8\alpha\tau^2}{R} = -\frac{(32\alpha-11)\tau^2}{3R}$$

where the terms give the macroion–rod–rod repulsion, the attraction between the macroion–rods and the adjacent ions on its outer side, the attraction of the ions on the outer side of the opposite macroion–rod and the attraction of the inner ions, in this order. For $\alpha \geq 11/32$ attraction occurs, while for $\alpha < 11/32$ the force is repulsive, i. e. the rods would drift apart, and the planar configuration cannot be a global minimum.

Stability of the particle–rods

The next thing to look at is the stability of the particle–rods. Therefore we determine the second order coefficient of the energy if we move a single particle out of the macroion–rod plane into the y–dimension (for symmetry reasons the first order

coefficient is zero). As the potential has a local minimum around both macroion–rods, the particle will stick to the rod surface, and move on a circle.

For the interaction of a single particle with a line of particles at distance $R\delta$ we obtain from the near formula (6.16)

$$\Delta E_{pp}(q, l, R; \delta) = \frac{q^2}{2l^3}\psi^{(2)}(1)R^2\delta^2 + \mathcal{O}(\delta^4)\,. \tag{7.42}$$

l denotes the charge separation of the particles within the line, i. e. $l = -q/(2\alpha\tau)$ for the inner particle–rod. The presented formula corresponds to the near formula for $\tilde{\psi}$, i. e. the line of particles has a gap at the position of the removed particle, as necessary. Note that $\psi^{(2)}(1) < 0$, so that this is a repulsive potential as expected.

For the interaction between a single particle, moving on a circle $R(\cos\delta - 1, \sin\delta)$ around $(-R, 0)$, and a rod at position $(d, 0)$, we obtain

$$\begin{aligned}\Delta E_{rp}(q, \tau, R, d; \delta) &= -q\tau \log \frac{(d + R\cos\delta - R)^2 + R^2\sin^2\delta}{d} \\ &= q\tau\frac{R(d-R)}{d^2}\delta^2 + \mathcal{O}(\delta^4)\,.\end{aligned} \tag{7.43}$$

Combining this, we obtain for an inner particle moving on the surface of the right macroion–rod an energy difference of

$$\begin{aligned}\Delta E_i = {} & \Delta E_{pp}(q, -q/(2\alpha\tau), R; \delta) + \Delta E_{rp}(q, \tau, R, R; \delta) + \Delta E_{rp}(q, \tau, R, -R; \delta) + \\ & \Delta E_{rp}(q, -(1-\alpha)\tau, R, 2R; \delta) + \Delta E_{rp}(q, -(1-\alpha)\tau, R, -2R; \delta) \\ = {} & -q\tau\left(2 - \frac{1}{2}(1-\alpha) + 4\alpha^3\gamma_z^{-2}\psi^{(2)}(1)\right)\delta^2 + \mathcal{O}(\delta^4)\,,\end{aligned} \tag{7.44}$$

where we introduced the dimensionless parameter

$$\gamma_z := -\frac{q}{\tau R} > 0\,, \tag{7.45}$$

which essentially measures the ratio between the average charge separation parallel to the rods and the rod radius, similar to the Rouzina–Bloomfield parameter. Large γ_z corresponds to a large distance of the charges within the particle–rod compared to their distance to the rods. Therefore one will expect stable particle–rods only for large γ_z, which will be shown rigorously now.

Since a configuration is only stable if $\delta = 0$ is a local minimum and since due to charge neutrality $-q\tau > 0$, we obtain an implicit upper limit for α from the condition

$$\left(\frac{3}{2} + \frac{1}{2}\alpha\right)\gamma_z^2 + 4\alpha^3\psi^{(2)}(1) > 0\,. \tag{7.46}$$

Inequality 7.46 is an upper bound on α, since $\psi^{(2)}(1)$ is negative, so that for sufficiently large α the second, negative term will dominate. Therefore $\alpha = 1$ is stable if and only if $\gamma_z > \sqrt{-2\psi^{(2)}(1)} =: \gamma_0 \approx 2.19$.

A similar calculation gives for the stability of the outer particle–rod

$$\Delta E_o = -q\tau\left(-\frac{2}{9} + \frac{3}{16}(1-\alpha) + \frac{\alpha}{2} + \frac{1}{2}(1-a)^3\gamma_z^{-2}\psi^{(2)}(1)\right)\,, \tag{7.47}$$

which gives an implicit lower limit on α from the condition

$$\frac{5}{16}\left(\alpha-\frac{1}{9}\right)\gamma_z^2-\frac{1}{2}(1-a)^3\psi^{(2)}(1)>0\,. \tag{7.48}$$

Therefore $\alpha \leq 1/9$ is never stable, i.e. not all ions can leave the gap between the macroion–rods, as one would have expected.

The energetically optimal α

Now that we know which for which values of α the planar configurations are stable, we compare their energies to determine the lowest energy state. To this aim we calculate the energy difference to the $\alpha = 1$ state of a finite size portion of the system along the z–axis of length λ_z.

For the energy between a rod and a particle we have

$$E_{rp}(q,\tau,d) = -2q\tau\left(\gamma+\log\frac{d}{2\lambda_z}\right).$$

The self energy of the particle–rods we can obtain from the MMM1D self energy. Let $N = \lambda_z/l$ be the number of particles of the rod per one box length. Then we would have to add up a lot of two particle interactions with $\rho = 0$ and varying γ_z, resulting in an intractable sum of polygamma terms. To avoid this, we use the self energy formula of MMM1D for a box length of l instead of λ_z to treat the interaction of a single particle with the complete rod at once. N–times this value then gives twice the self energy of the particle–rod per box length λ_z. We obtain the self energy

$$\frac{1}{2}N\left(-\frac{2q^2}{l}\log(\beta l)\right)+\mathcal{O}(\beta)=\frac{\lambda_z q^2}{l^2}\left(-\log(\beta\lambda_z)+\log\frac{\lambda_z}{l}\right)+\mathcal{O}(\beta)$$

with the correct singularity $-N^2/2q^2/\lambda_z\log(\beta\lambda_z)$. In the limit $\beta\to 0$ we obtain

$$E_{pp}(q,l)=\frac{\lambda_z q^2}{l^2}\log\frac{\lambda_z}{l}\,. \tag{7.49}$$

We obtain the total energy difference to the $\alpha = 1$ configuration as

$$\begin{aligned}
\Delta E &= E_{pp}(q,-q/(2\alpha\tau))+2E_{pp}(q,-q/((1-\alpha)\tau))-E_{pp}(q,-q/(2\tau))\\
&\quad+\frac{2\lambda_z}{-q/(2\alpha\tau)}\left(E_{rp}(q,\tau,R)+2E_{rp}(q,-(1-\alpha)\tau,2R]\right)-\frac{2\lambda_z}{-q/(2\tau)}E_{rp}(q,\tau,R)\\
&\quad+\frac{2\lambda_z}{-q/((1-\alpha)\tau)}\left(E_{rp}(q,\tau,R)+E_{rp}(q,\tau,3R)\right)\\
&\quad+\frac{\lambda_z}{-q/((1-\alpha)\tau)}E_{rp}(q,-(1-\alpha)\tau,4R)\\
&=2\tau^2\lambda_z\Bigg(-\gamma+\log\frac{9}{2}-\left(2\gamma+\log\frac{9}{4}\right)\alpha+(3\gamma-\log 2)\,\alpha^2\\
&\qquad+(1-\alpha)(1+3\alpha)\log\frac{\lambda_z}{R}-2\log\left(-\frac{2\lambda_z\tau}{q}\right)\\
&\qquad+(1-\alpha)^2\log\frac{-(1-\alpha)\lambda_z\tau}{q}+2\alpha^2\log\frac{-2\alpha\lambda_z\tau}{q}\Bigg).
\end{aligned}$$

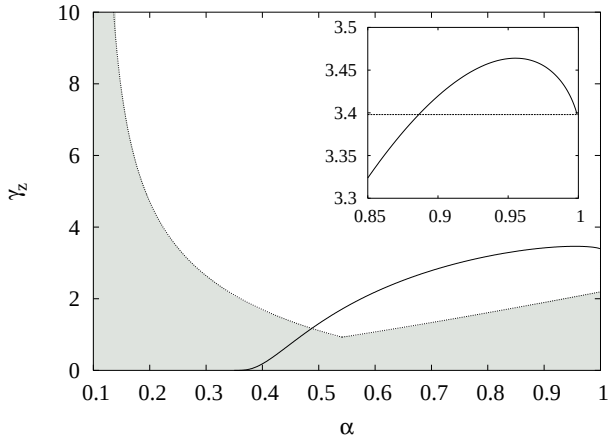


Figure 7.14: Theoretical predictions for α and γ_z. The grey area marks the values for α for which either the inner particle line is instable (right of peak) or the outer one (left of the peak). The solid line marks the local extremum of $\Delta E(\alpha, \gamma_z)$ with respect to α. On the inset one can see that each $\gamma_z > \gamma_z^{split}$ (dotted line) is for two values of α a local minimum.

The sum of the prefactors of the logarithmic terms is zero. Therefore we can multiply in a constant factor $-q/(\tau\lambda_z)$ inside the logarithms and obtain for the energy difference per length

$$\Delta E/\lambda_z = 2\tau^2\Big(-\gamma + \log\frac{9}{8} - \Big(2\gamma + \log\frac{9}{4}\Big)\,\alpha + (3\gamma + \log 2)\,\alpha^2 + (1-\alpha)^2\log(1-\alpha) + 2\alpha^2\log(\alpha) + (1-\alpha)(1+3\alpha)\log(\gamma_z)\Big). \tag{7.50}$$

We are interested in the minimum of this function with respect to $1/9 < \alpha \leq 1$, which is either at $\alpha = 1$ with an energy difference of 0 or where

$$0 = \partial_\alpha \Delta E/\lambda_z = -2\tau^2\Big(-1 - 2\gamma - \log\frac{9}{4} + \alpha\,(3 + 6\gamma + \log 2) - 2(1-\alpha)\log(1-\alpha) + 4\alpha\log\alpha + (2-6\alpha)\log(\gamma_z)\Big).$$

This equation can be solved easily for γ_z, which gives

$$\gamma_z = e^{\frac{1}{6\alpha-2}\left(-1-2\gamma-\log\frac{9}{4}+\alpha(3+6\gamma+\log 2)-2(1-\alpha)\log(1-\alpha)+4\alpha\log\alpha\right)}. \tag{7.51}$$

Obviously $\gamma_z(\alpha)$ has a divergence at $\alpha = 1/3$. The part not dependent on γ_z has a negative sign for $\alpha < 0.4$, so that for $\alpha \to 1/3_+$, the exponent diverges to $-\infty$, while γ_z tends to 0. Fig. 7.14 shows $\alpha(\gamma_z)$ for α between 1/3 and 1, Fig. 7.15 the corresponding energy difference at the free minimum. Values of α below 1/3 are unimportant, since for these the energy at the local minimum is positive, i. e. larger than for $\alpha = 1$, and therefore cannot be global minima.

Fig. 7.15 shows that the energy difference is smaller than zero only for $\alpha < \alpha_{max} \approx 0.927$. Therefore larger values of α do not occur as local minima. However, $\alpha = 1$ can occur as global boundary minimum.

Looking at the local minima with respect to γ_z, the situation is more complicated. Each $\gamma_z > \gamma_z^{split} \approx 3.398$ is a local minimum for two values of α, see Fig. 7.14. However, the larger of these α values is always larger than α_{max}, and therefore cannot be the

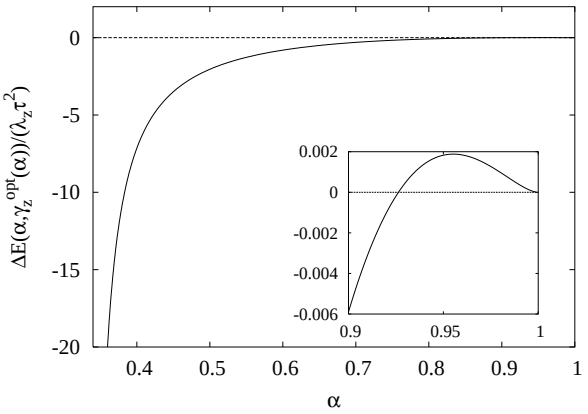


Figure 7.15: Energy difference $\Delta E/(\lambda_z \tau^2)$ in the local extremum γ_z^{opt} as shown in Fig. 7.14. For values larger than 0, $\alpha = 1$ is energetically favourable. Therefore with increasing γ_z the global minimum grows until it reaches α_{max} and jumps up to 1.

global minimum. But even the lower one of these α values is larger than α_{max} for $\gamma_z > \gamma_z^{crit} \approx 3.451$, so that for these γ_z values there is no local minimum, and therefore $\alpha = 1$ is the global minimum.

Below $\gamma_z = \gamma_z^{min} \approx 1.175$ the free minimum lies outside of the stability region for the planar configuration. Therefore it is energetically more favourable for particles of the outer particle–rod to leave the rod–rod plane and sit somewhere else on the macroion–rod surfaces.

We obtain the following picture of the system configurations for different values of γ_z:

- For $\gamma_z \geq \gamma_z^{crit}$, the optimal α is one, so that the particles are lined up in the centre of the gap between the rods, and the strong coupling picture holds.
- For γ_z below γ_z^{crit}, but above γ_z^{min}, the optimal α value is smaller than one, however, the particles are still located in the rod–rod plane. The particles arrange in two counterion–rods on the outside and one counterion–rod on the inside.
- Below γ_z^{min}, the outer counterion–rod breaks up first, while the inner particle rod is still stable.
- For very small γ_z, even the inner particle–rod becomes instable, and the particles distribute nearly regularly on the rod surfaces.

These qualitatively different groundstate types are sketched in Fig. 7.16.

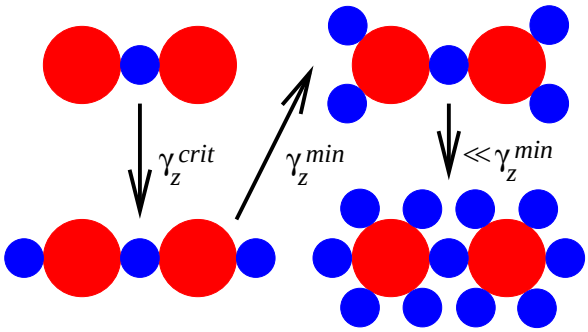


Figure 7.16: Cartoon of the different groundstate types developed with decreasing γ_z. particles are represented by blue circles, the macroion–rods by red ones.

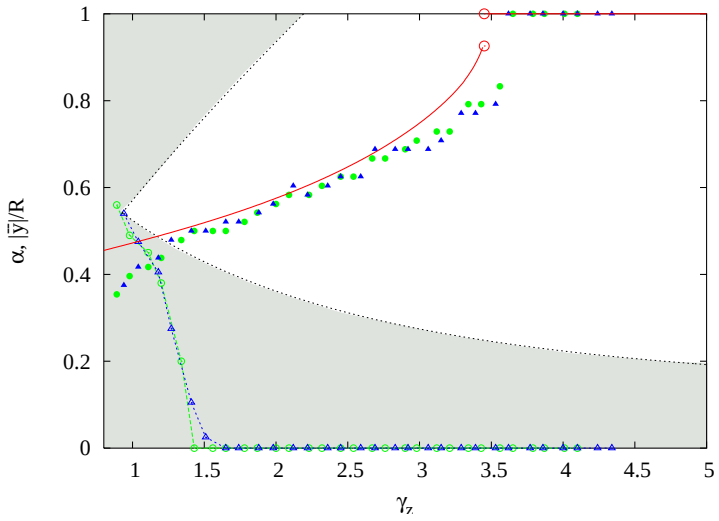


Figure 7.17: Simulation results for two simulation runs with $R = 1$ and $R = 2$. Again the grey area denotes values of α for which the particle rods get instable; below the bottom line the outer particle–rods are instable, above the top line the inner rod. The filled squares denote the α value of the approximated ground state as a function of γ_z for $R = 1$, the filled triangles for $R = 2$. The open symbols show the average distance $\bar{y}$ of the particles to the rod–rod plane in multiples of the rod radius R.

Note that both the constraints for possible values of α obtained from the stability considerations for the particle–rods and the position of the free minimum of the energy only depend on α and γ_z. Since α is determined by optimisation, the only parameter left in the formulas is γ_z, which alone determines the behaviour of the system. For $\gamma_z > \gamma_z^{crit}$ the system should behave according to the strong coupling theory, while for smaller γ_z a planar behaviour is predicted. The transition from the strong coupling configurations to the planar configurations is discontinuous, particle fractions between α_{max} and 1 cannot be obtained as ground states. For $\gamma_z < \gamma_z^{min}$, the planar configurations are no longer stable, and three dimensional configurations are energetically more favourable.

7.6 Comparison with numerical results

To determine the ground state numerically, MD simulations of a system of 48 counterions and 2 rods were done. The particles were placed initially in particle rods with $\alpha = 0.5$ as described above. Then the system was continously heated up and cooled down again, until the energy gain between two cycles was smaller than 0.1%. The peak temperature was $1kT$ for the first cycle and reduced by a factor of 0.5 for every following cycle, each cycle consists of 80000 integrator steps. As ground state the configuration with the lowest potential energy is used. In addition to these configurations, the $\alpha = 1$ configuration was generated manually and used if the energy was lower than for the simulation configurations.

Note that for the hard core interactions between the rods and the counterions

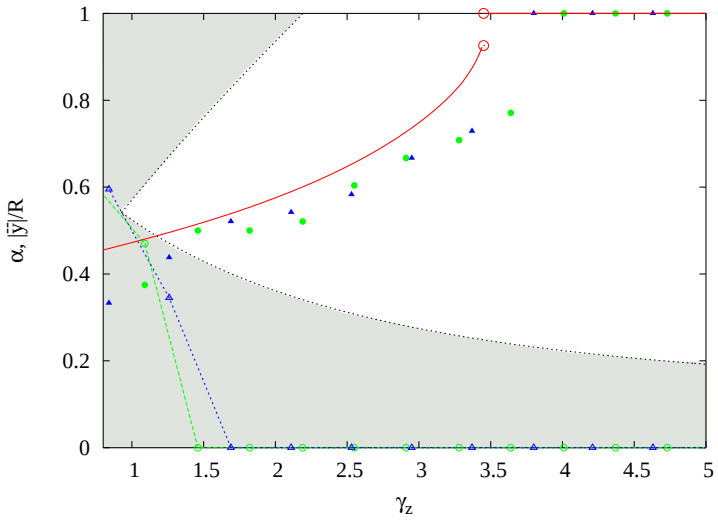


Figure 7.18: The same as Fig. 7.17, but for $\Delta = 0.5$. Here $\gamma_z = -q/(\tau(R + \Delta/2))$ was used (see text).

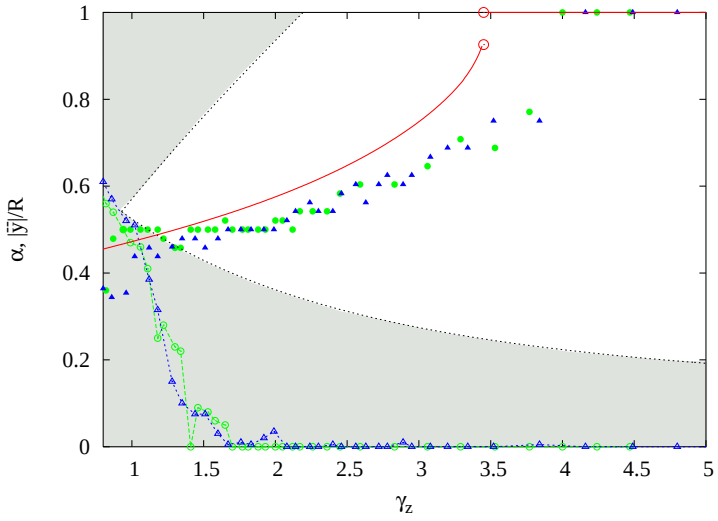


Figure 7.19: The same as Fig. 7.17, but for $\Delta = 2$. Here $\gamma_z = -q/(\tau(R + \Delta/2))$ was used (see text). Note that the values below 1.5 are not meaningful; as one can see e. g. in Fig. 7.24, most of the particles are located close to the dividing plane between inner and outer ions, so that α cannot be measured reasonably from the simulation.

again a purely repulsive Lennard–Jones potential was used (see Sec. 1.5). To be able to obtain the flat configurations described above, there has to be a line between the rods on which the soft repulsion potential does not act on the particles. Therefore in the simulations the gap size was increased slightly by 0.245.

Due to the fixed number of particles, only regular patterns consisting of $1, 2, 3, 4, 6, \ldots, 48$ particles can be formed. Therefore ground states patterns with for example two counterions on each outer rod and three counterions on the inner rod are not accessible, which may lead to frustration. However, the system energy is dominated by the rod–rod interaction both of the macroion– and counterion–rods, so that the placing of the particles within the particle rods only plays a minor role. Therefore the obtained optimal value of α is close to the correct value, although the particle distribution may not have the correct regular pattern. The mentioned seven–particle–pattern from above is for example approximated by a mix of six and eight particle patterns.

Fig. 7.17 shows the optimal α from two simulation runs with $q = 3$ and $R = 1$ rsp. $R = 2$; τ was calculated from γ_z via $\tau = -q/(R\gamma_z)$. One can see that the approximated ground state configurations obtained from the simulation for $\gamma_z > 1.6$ are indeed flat as assumed in the previous section. The particles leave the rod–rod plane when for the optimal α the flat configuration is no longer stable. Interestingly, although the assumptions for the theoretical calculations are no longer valid for $\gamma_z < 1.6$, the predictions for α still fit quite well. This shows again that the total energy of the system is determined primarily by the counterion fractions in the gap between the rods and outside of it, and not as much by the exact location of the counterions.

Note that the obtained α value is slightly lower than the theoretical prediction, especially for α between 0.7 and 1, which can be understood as follows:

The only approximation in the theoretical calculations is the approximation of the particle–particle interactions for different particle rods by a continuous rod. For real particles one can see from the far formula of MMM1D that for an optimal placement the energy will be lower than for a rod. Therefore the repulsion induced by the particle–particle interactions between different particle rods is lower than the one obtained from the rod model, so that the real α will be closer to $\alpha = 1/2$. Moreover, the slightly increased gap size also allows the particles in the gap to place further away from each other by adopting a zig–zag pattern, so that the self–energy of the inner counterion–rod is lower than for $\Delta = 0$. Of course this argument does not hold for $\alpha = 1$, since in this case the approximations are exact.

Fig. 7.18 and Fig. 7.19 show the same as Fig. 7.17, but this time for a gap size larger than zero, namely of half the rod radius and twice the rod radius. The radii are again increased by 0.245 for comparability. One can see that the optimal value of α drops further, which supports the arguments of the previous paragraph. Nevertheless the flat configurations still seem to form the ground states as long as they are stable. Especially the $\Delta = 2$ case is far from the assumptions made in the previous section; remarkably though, the optimal α differs only by at most 30% from results for $\Delta = 0$.

Figs 7.20 through 7.24 show snapshots of the above mentioned simulations from all the different groundstate types discussed in the previous section:

- Fig. 7.20 shows a groundstate for $\gamma_z \geq \gamma_z^{crit}$, for which the strong coupling picture holds.

- For Figs. 7.21 and 7.22, $\gamma_z^{min} < \gamma_z < \gamma_z^{crit}$. The particles are organised in particle–rods, in the case of both figures with $\alpha = 1/2$, however at different surface to surface distances.
- Fig. 7.23 shows a snapshot for which $\gamma_z < \gamma_z^{min}$, so that the outer counterion–rod has split up. However, the inner particle rod is still stable.
- Fig. 7.24 finally shows a snapshot where the particles are distributed nearly regularly around both rod surfaces.

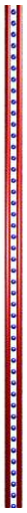


Figure 7.20: Snapshot of a simulated ground state for $\Delta = 0$,$R = 2$ and $\gamma_z = 4.60$, i. e. in the SC regime. Note that the counterions and rods are shown much smaller than they really are for better visibility.

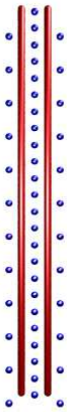


Figure 7.21: Snapshot of a simulated ground state for $\Delta = 0$,$R = 1$ and $\gamma_z = 1.6$, i. e. below γ_z^{crit}. The charge fractions inside and outside are equal, so that $\alpha = 1/2$.

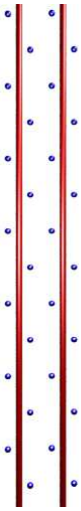


Figure 7.22: Snapshot of a simulated ground state for $\Delta = 0.5$,$R = 1$ and $\gamma_z = 4$, again with $\alpha = 1/2$. Due to the larger surface to surface distance, the configuration differs from Fig. 7.21.

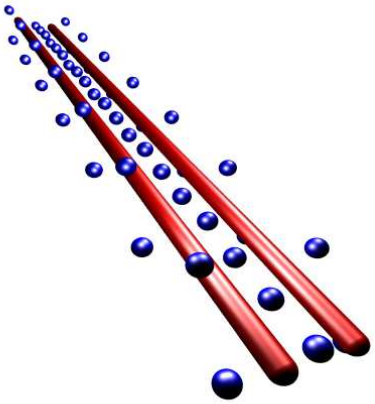


Figure 7.23: Snapshot of a simulated ground state for $\Delta = 0$,$R = 1$ and $\gamma_z = 1.25$, where the outer particle–rods have broken apart, however the inner particle rods are still stable.

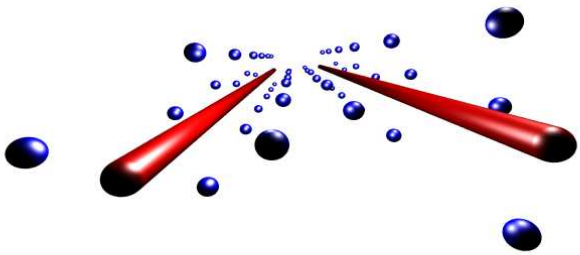


Figure 7.24: Snapshot of a simulated ground state for $\Delta = 2$,$R = 1$ and $\gamma_z = 1$. Here the particles are spread equally on the surfaces of the rods.

8 Conclusions

The simulation package ESPResSo presented at the beginning of this work provides an extensible, reliable and computationally efficient framework for MD simulations that makes the implementation of new algorithms as easy as possible. The computational efficiency of ESPResSo is primarily due to its flexible, but efficient particle data organisation scheme.

Two methods for the calculation of the electrostatic sum in two dimensionally periodic systems have been implemented in ESPResSo, MMM2D and ELC. MMM2D has been parallelised for the implementation in ESPResSo, achieving a good efficiency and a minimal scaling of $\mathcal{O}(N^{4/3})$. On a single processor, MMM2D can be used efficiently for up to 400 particles.

The ELC correction term derived in this work allows to use methods for three dimensionally periodic systems also for the case of only two periodic coordinates. This is achieved by replicating the system periodically along the nonperiodic coordinate and correcting for the additional contributions again, which can be done in linear computation time. Currently ELC is the fastest method for two dimensionally periodic systems with less than 10^6 particles which features a rigorous error bound.

In this work it has been described how MMM2D and ELC can be modified to treat different dielectric constants inside and outside of the simulation box, as one will typically have in the simulation of a thin film of water or a metallic surface.

Starting from MMM2D, MMM1D for systems with only one periodic coordinate was developed. MMM1D is the fastest method for one dimensionally periodic systems with a rigorous error bound. The computation time of MMM2D scales quadratically with the number of particles, but the prefactor is small enough to allow efficient simulations with up to 400 particles.

Using MMM1D, the problem of the attraction of two like–charged rods was addressed. Results of computer simulations showed a good agreement with the strong–coupling theory of R. Netz. As a measure of agreement between the simulation at finite coupling parameter and the theory for infinite coupling parameter, the Rouzina–Bloomfield parameter was confirmed. The Rouzina–Bloomfield parameter is a measure for the importance of counterion–counterion correlations, which are ignored in the strong–coupling theory.

At zero temperature, the SC theory predicts a zero surface to surface distance of the rods, with the particles lined up in the gap between the rods. While the zero surface to surface distance can be confirmed by computer simulations, the particle distributions shows a more complex behaviour. For certain parameters planar configurations occur, for which the stability was determined and an analytic approximation of the potential energy was derived. Both the stability and the minimum of the energy are described by a single parameter γ_z similar to the Rouzina–Bloomfield parameter. These results were confirmed by computer simulations, even at comparatively large surface to surface distances.

Contents of the attached CD

The attached compact disc contains the sources of Espresso in the 1.6.1d version and the scripts used to generate the data presented in this work, plus two example simulation directories. The disc is formatted according to the ISO9660 format with Joliet and Rockridge extensions.

ESPResSo

The attached version 1.6.1d is the most current version of ESPResSo that was used to produce the data presented in this work. It resides in the directory *Espresso*. The HTML documentation of Espresso can be found in *Espresso/doc/html*, the latex documentation in *Espresso/doc/latex/refman.dvi*. However, the HTML documentation is much more comfortable and allows to traverse from a Tcl–command to its source code with a few mouse clicks.

To build Espresso, the following software has to be installed:

MPI: For GNU/Linux and MacOS the LAM/MPI is recommended, on AIX and Tru64 you can use the native MPI library. LAM/MPI is available via the URL `http://www.lam-mpi.org/`.

Tcl/Tk **8.4 or newer**: Tcl/Tk is needed on all platforms and available via the URL `http://tcl.activestate.com`. Most modern operating systems contain a sufficiently recent version of Tcl.

FFTW 2.1.x: FFTW is needed on all platforms and available via the URL `http://www.fftw.org/`.

For details on installing and testing the ESPResSo installation, see the *README* file or the HTML documentation on the disc. Additional information can be found via the ESPResSo web page `http://www.espresso.mpg.de`.

Scripts

The scripts specific to this thesis reside in the *scripts* directory. The directory *scripts/timings_and_errors* contains the scripts used to test the performance of MMM1D, MMM2D and ELC. The `sample_*` scripts were used to generate the error distributions, the `timing_*` scripts the computation times.

The scripts in *scripts/two_rod_problem* were used to produce and analyse the data for chapter 7. In the directory a *README* file is included that contains details on the use of the scripts. Sample data can be found in the *simulations* subdirectory.

The *biro_diss.tcl* script is the simulation script which performs the bisection algorithm, *biro_diss_redo.tcl* generates the data for a Newton step. After the simulation, the data has to be condensed into a packed trajectory format by the script *repack.tcl*. *interpol.tcl* uses this trajectory format and calculates the improved zero guess by linear interpolation. The results are written into *result.log*, which is used by *biro_diss_redo.tcl* to determine its sampling points. Therefore one can simple start *biro_diss_redo.tcl* several times to obtain several Newton steps.

The particle distributions were generated using the *calc_ion_distribution.tcl* script. For details on the usage, see the comments at the beginning of the scripts.

Acknowledgements

I thank my family for their support and their passion with me during the creation of this thesis.

Curriculum vitae

Personal data

Axel Arnold
Hofheimer Weg 18
63110 Rodgau
born September 4, 1975 in Ulm, Germany

Education

1982–1985	Grundschule Wesseling
1985–1986	Grundschule Rodgau–Niederroden
1986–1988	Förderstufe Rodgau–Jügesheim
1988–1995	Leibniz-Gymnasium Offenbach
June 28, 1995	Abitur
1995-1996	military service
1996-2001	studies of mathematics and computer science at the Johannes–Gutenberg–Universität Mainz, Germany
October 2, 2001	diploma of mathematics
since 2001	doctorate at the Max–Planck–Institut für Polymerforschung, Mainz, Germany

Bibliography

[1] M. Abramowitz and I. Stegun. *Handbook of mathematical functions*. Dover Publications Inc., New York, 1970.

[2] T. Alfrey, P. W. Berg, and H. J. Morawetz. The counterion distribution in solutions of rod-shaped polyelectrolytes. *J. Polym. Sci.*, 7:543, 1951.

[3] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford Science Publications. Clarendon Press, Oxford, 1 edition, 1987.

[4] A. Arnold. Berechnung der elektrostatischen wechselwirkung in 2d + h periodischen systemen. Diploma thesis, Johannes Gutenberg-Universität, may 2001.

[5] A. Arnold, J. de Joannis, and C. Holm. Electrostatics in periodic slab geometries i. *J. Chem. Phys.*, 117:2496–2502, 2002.

[6] A. Arnold and C. Holm. Mmm2d: A fast and accurate summation method for electrostatic interactions in 2d slab geometries. *Comp. Phys. Comm.*, 148(3):327–348, 1 November 2002.

[7] A. Arnold and C. Holm. A novel method for calculating electrostatic interactions in 2d periodic slab geometries. *Chemical Physical Letters*, 354:324–330, 2002.

[8] A. Arnold, B. Mann, H. J. Limbach, and C. Holm. ESPResSo — An Extensible Simulation Package for Research on Soft Matter Systems. In *Billing–Preis 2003*, 2003.

[9] A. Arnold, B. A. Mann, H. J. Limbach, and C. Holm. ESPResSo - An Extensible Simulation Package for Research on Soft Matter Systems. to be submitted.

[10] M. C. Barbosa, M. Deserno, and C. Holm. Ion-ion correlations: An improved one-component plasma correction. *Europhys. Lett.*, 52(1):80–86, 2000.

[11] J. Barnes and P. Hut. A hierarchical o(n-log-n) force calculation algorithm. *Nature*, 324:446, 1986.

[12] V. A. Bloomfield. DNA condensation. *Current Opin. Struct. Biol.*, 6:334, 1996.

[13] N. Boghossian, O. Kohlbacher, and H.-P.Lenhof. Ball: Biochemical algorithm library. Research report, Max-Planck-Institut für Informatik, Saarbrücken, 1999.

[14] CVS. Concurrent versions system - homepage, 2003. http://www.cvshome.org/.

[15] T. Darden, D. York, and L. Pedersen. Particle mesh ewald: An $n \log(n)$ method for ewald sums in large systems. *J. Chem. Phys.*, 98:10089, 1993.

[16] J. de Joannis, A. Arnold, and C. Holm. Electrostatics in periodic slab geometries ii. *J. Chem. Phys.*, 117:2503–2512, 2002.

[17] M. Deserno. *Counterion condensation for rigid linear polyelectrolytes*. PhD thesis, Universität Mainz, 2000.

[18] M. Deserno, A. Arnold, and C. Holm. Attraction and ionic correlations between charged stiff polyelectrolytes. *Macromolecules*, 36(1):249–259, 2003.

[19] M. Deserno and C. Holm. How to mesh up Ewald sums. i. a theoretical and numerical comparison of various particle mesh routines. *J. Chem. Phys.*, 109:7678, 1998.

[20] M. Deserno and C. Holm. How to mesh up Ewald sums. ii. an accurate error estimate for the p3m algorithm. *J. Chem. Phys.*, 109:7694, 1998.

[21] M. Deserno and C. Holm. Cell-model and poisson-boltzmann-theory: A brief introduction. In C. Holm, P. Kékicheff, and R. Podgornik, editors, *Electrostatic Effects in Soft Matter and Biophysics*, volume 46 of *NATO Science Series II - Mathematics, Physics and Chemistry*. Kluwer Academic Publishers, Dordrecht, Nl, December 2001.

[22] M. Deserno, C. Holm, and S. May. The fraction of condensed counterions around a charged rod: Comparison of Poisson-Boltzmann theory and computer simulations. *Macromolecules*, 33:199–206, 2000.

[23] M. Deserno, F. Jiménez-Ángeles, C. Holm, and M. Lozada-Cassou. Overcharging of dna in the presence of salt: Theory and simulation. *Journal Phys. Chem. B*, 105(44):10983 –10991, October 2001.

[24] M. Doi. Octa homepage, 2003. http://octa.jp/OCTA/whatsOCTA.html.

[25] B. Dünweg and W. Paul. Brownian dynamics simulations without gaussian random numbers. *Int. J. Mod. Phys. C*, 2:817, 1991.

[26] W. Earnshaw. On the nature of the molecular forces which regulate the constitution of the luminferous ether. *Trans. Camb. Phil. Soc.*, 7:97–112, 1842.

[27] ESPResSo. Homepage, 2004. http://www.espresso.mpg.de.

[28] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. Pedersen. A smooth particle mesh ewald method. *J. Chem. Phys.*, 103:8577, 1995.

[29] P. Ewald. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.*, 64:253–287, 1921.

[30] FFTW. Homepage, 2003. http://www.fftw.org/.

[31] D. Frenkel and B. Smit. *Understanding Molecular Simulation*. Academic Press, San Diego, second edition, 2002.

[32] R. M. Fuoss, A. Katchalsky, and S. Lifson. The potential of an infinite rod-like molecule and the distribution of the counter ions. *Proc. Natl. Acad. Sci. USA*, 37:579–589, 1951.

[33] W. M. Gelbart, R. F. Bruinsma, P. A. Pincus, and V. A. Parsegian. Dna-inspired electrostatics. *Physics Today*, 53(9):38–44, 2000.

[34] E. González-Tovar and M. Lozada-Cassou. Hypernetted chain approximation for the distribution of ions arround a cylindrical electrode .2. numerical-solution for a model cylindrical poly-electrolyte. *J. Chem. Phys.*, 83:361–370, 1985.

[35] G. L. Gouy. *J. de Phys.*, 9:457, 1910.

[36] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997.

[37] G. S. Grest, B. Dünweg, and K. Kremer. Vectorized link cell fortran code for molecular-dynamics simulations for a large number of particles. *Comp. Phys. Comm.*, 55(3):269–285, 1989.

[38] G. S. Grest and K. Kremer. Molecular dynamics simulation for polymers in the presence of a heat bath. *Phys. Rev. A*, 33(5):3628–31, 1986.

[39] N. Grønbech-Jensen, R. J. Mashl, R. F. Bruinsma, and W. M. Gelbart. Counterion-induced attraction between rigid polyelectrolytes. *Phys. Rev. Lett.*, 78:2477–2480, 1997.

[40] L. Guldbrand, L. G. Nilsson, and L. Nordenskiöld. A monte carlo simulation study of electrostatic forces between hexagonally packed dna double helices. *J. Chem. Phys.*, 90:5893, 1989.

[41] L. Guldbrand, L. G. Nilsson, and L. Nordenskiöld. Evaluation of the electrostatic osmotic pressure in an infinite system of hexagonally oriented dna molecules — a monte–carlo simulation study. *MPhys*, 72:177–192, 1991.

[42] B.-Y. Ha and A. J. Liu. Counterion-mediated attraction between two like-charged rods. *Phys. Rev. Lett.*, 79(7):1289, Aug. 1997.

[43] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. IOP, London, 1988.

[44] C. Holm, P. Kékicheff, and R. Podgornik, editors. *Electrostatic Effects in Soft Matter and Biophysics*, volume 46 of *NATO Science Series II - Mathematics, Physics and Chemistry*. Kluwer Academic Publishers, Dordrecht, Nl, December 2001.

[45] W. Humphrey, A. Dalke, and K. Schulten. VMD: Visual molecular dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.

[46] R. A. L. Jones. *Soft Condensed Matter*. Oxford University Press, Great Clarendon Street, Oxford OX2 6DP, 2002.

[47] M. Kawata and U. Nagashima. Particle mesh ewald method for three-dimensional systems with two-dimensional periodicity. *Chem. Phys. Lett.*, 340:165–172, 2001.

[48] J. Kolafa and J. W. Perram. Cutoff errors in the ewald summation formulae for point charge systems. *Molecular Simulation*, 9(5):351–68, 1992.

[49] A. A. Kornyshev and S. Leikin. Theory of interaction between helical molecules. *J. Chem. Phys.*, 108:7035, 1998.

[50] O. Lambert, L. Letellier, W. Gelbart, and J. Rigaud. Dna delivery by phage as a strategy for encapsulating toroidal condensates of arbitrary size into liposomes. *Proc. Natl. Acad. Sci. (USA)*, 97(13):7248–7253, 2000.

[51] LAM/MPI. Homepage, 2004. http://www.lam-mpi.org/.

[52] M. Le Bret and B. Zimm. Distribution of counterions around a cylindrical polyelectrolyte and manning's condensation theory. *Biopolymers*, 23:287, 1984.

[53] M. Le Bret and B. Zimm. Monte carlo determination of the distribution of ions about a cylindrical polyelectrolyte. *Biopolymers*, 23:271, 1984.

[54] C. J. Lejdfors. Gismos home page, 1998. http://www.teokem.lu.se/gismos/.

[55] J. Lekner. Summation of dipolar fields in simulated liquid vapor interfaces. *Physica A*, 157:826, 1989.

[56] L.Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* MIT Press, Cambridge, MA, 1988.

[57] L.Greengard and V. Rhoklin. A fast algorithm for particle simulations. *J. Comp. Phys.*, 73:325, 1987.

[58] A. P. Lyubartsev and L. Nordenskiöld. Monte carlo simulation study of dna polyelectrolyte properties in the presence of multivalent polyamine ions. *J. Phys. Chem.*, 101:4335, 1997.

[59] A. P. Lyubartsev, J. X. Tang, P. A. Janmey, and L. Nordenskiöld. Electrostatically induced polyelectrolyte association of rodlike virus particles. *Phys. Rev. Lett.*, 81(24):5465, Dec. 1998.

[60] M. M. Lekner summations. *J. Chem. Phys*, 115(7):2955, Aug 2001.

[61] A. Maggs and V. Rosseto. Local simulation algorithms for coulombic interactions. *Phys. Rev. Lett.*, 88:196402, 2002.

[62] G. Manning. Limiting laws and counterion condensation in polyelectrolyte solutions i. colligative properties. *J. Chem. Phys.*, 51:924–933, 1969.

[63] G. Manning. limiting laws and counterion condensation in polyelectrolyte solutions ii. self-diffusion of the small ions. *J. Chem. Phys.*, 51:934–938, 1969.

[64] G. Manning. limiting laws and counterion condensation in polyelectrolyte solutions iii. an analysis based on the mayer ionic solution theory. *J. Chem. Phys.*, 51:3249, 1969.

[65] A. G. Moreira and R. R. Netz. Binding of similarly charged plates with counterions only. *Phys. Rev. Lett.*, 87:078301, 2001.

[66] A. G. Moreira and R. R. Netz. Field–theoretic approaches to classical charged systems. In C. Holm, P. Kékicheff, and R. Podgornik, editors, *Electrostatic Effects in Soft Matter and Biophysics*, volume 46 of *NATO Science Series II - Mathematics, Physics and Chemistry*. Kluwer Academic Publishers, Dordrecht, Nl, December 2001.

[67] A. G. Moreira and R. R. Netz. Simulations of counterions at charged plates. *Eur. Phys. J. E*, 8(1):33–58, 2002.

[68] MPICH. Homepage, 2004. http://www-unix.mcs.anl.gov/mpi/mpich/.

[69] A. Naji, A. Arnold, C. Holm, and R. R. Netz. Attraction and unbinding of like–charged rods. *Europhys. Lett.*, 67:130–136, 2004.

[70] A. Naji and R. Netz. Attraction of like–charged macroions in the strong coupling limit. *Eur. Phys. J. E*, 13(1):43–59, 2004.

[71] M. Nelson, W. Humphrey, A. Gursoy, A. Dalke, L. Kale, R. Skeel, and K. Schulten. Namd– a parallel, object-oriented molecular dynamics program. *Int. J. Supercomput. Ap.*, 10(4):251–268, 1996.

[72] J. Neu. Wall-mediated forces between like-charged bodies in an electrolyte. *Phys. Rev. Lett.*, 82(5):1072–1074, 1999.

[73] B. R. A. Nijboer and F. W. de Wette. On the calculation of lattice sums. *Physica*, 23:309–321, 1957.

[74] F. Oberhettinger. *Fouriertransforms of distributions and their inverses*, volume 16 of *Probability and Mathematical Statistics*. Academic Press, New York, 1973.

[75] F. Oded. "water free" computer model for lipid bilayer membranes. *J. Chem. Phys.*, 119(1):596–605, 2003.

[76] F. Oosawa. *Polyelectrolytes*. Marcel Dekker, New York, 1971.

[77] D. Pearlman, D. Case, J. Caldwell, W. Ross, I. T.E. Cheatham, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman. AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Comp. Phys. Comm*, 91:1–41, 1995.

[78] H. G. Petersen. Accuracy and efficiency of the particle mesh ewald method. *J. Chem. Phys.*, 103(9):3668–79, 1995.

[79] S. J. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comp. Phys.*, 117:1–19, 1995.

[80] R. Podgornik, D. Rau, and A. Parsegian. Parametrization of direct and soft steric-undulatory forces between dna double-helical polyelectrolytes in solutions of several different anions and cations. *Biophys. J.*, 66:962, 1994.

[81] M. Porto. Ewald summation of electrostatic interactions of systems with finite extent in two of three dimensions. *J. Phys. A*, 33:6211, 2000.

[82] M. Pütz and A. Kolb. Optimization techniques for parallel molecular dynamics using domain decomposition. *Comp. Phys. Comm.*, 113:145–167, 1998.

[83] J. Ray and G. S. Manning. Effect of counterion valence and polymer charge density on the pair potential of two polyions. *Macromolecules*, 30(19):5739–5744, 1997.

[84] I. Rouzina and V. Bloomfield. Macroion attraction due to electrostatic correlation between screening counterions. 1. mobile surface-adsorbed ions and diffuse ion cloud. *J. Phys. Chem.*, 100(23):9977–9989, 1996.

[85] J. Sader and D. Y. Chan. Long-range electrostatic attractions between identically charged particles in confined geometries: An unresolved problem. *J. Colloid Interface Sci.*, 213:268, 1999.

[86] B. I. Shklovskii. Wigner crystal model of counterion induced bundle formation of rodlike polyelectrolytes. *Phys. Rev. Lett.*, 82(16):3268–3271, 1999.

[87] E. R. Smith. Electrostatic energy in ionic crystals. *Proc. R. Soc. Lond. A*, 375:475–505, 1981.

[88] E. R. Smith. Electrostatic potentials for thin layers. *Mol. Phys.*, 65:1089–1104, 1988.

[89] W. Smith and T. Forester. Dl_poly_2.0: A general-purpose parallel molecular dynamics simulation package. *J. Molec. Graphics*, 14:136, 1996.

[90] R. Sperb. Extension and simple proof of lekner's summation formula for coulomb forces. *Molecular Simulation*, 13:189–193, 1994.

[91] T. Straatsma, E. Aprà, T. Windus, E. Bylaska, W. de Jong, S. Hirata, M. Valiev, M. T. Hackler, L. Pollack, R. J. Harrison, M. Dupuis, D. Smith, J. Nieplocha, T. V., M. Krishnan, A. A. Auer, E. Brown, G. Cisneros, G. I. Fann, H. Fruchtl, J. Garza, K. Hirao, R. Kendall, J. Nichols, K. Tsemekhman, K. Wolinski, J. Anchell, D. Bernholdt, P. Borowski, T. Clark, D. Clerc, H. Dachsel, M. Deegan, K. Dyall, D. Elwood, E. Glendening, M. Gutowski, A. Hess, J. Jaffe, B. Johnson, J. Ju, R. Kobayashi, R. Kutteh, Z. Lin, R. Littlefield, X. Long, B. Meng, T. Nakajima, S. Niu, M. Rosing, G. Sandrone, M. Stave, H. Taylor, G. Thomas, J. van Lenthe, A. Wong, and Z. Zhang. *NWChem, A Computational Chemistry Package for Parallel Computers, Version 4.6.* Pacific Northwest National Laboratory, Richland, Washington 99352-0999, USA, 2004.

[92] R. Strebel. Pieces of software for the Coulombic m body problem. Dissertation 13504, ETH Zuerich, 1999.

[93] Tcl/Tk. Homepage, 2003. http://tcl.activestate.com/.

[94] E. Trizac. Effective interactions between like-charged macromolecules. *Phys. Rev. E*, 62(2):R1465–R1468, 2000.

[95] E. Trizac and J.-L. Raimbault. Long-range electrostatic interactions between like-charged colloids: steric and confinement effects. *Phys. Rev. E*, 60:6530, 1999.

[96] I. Tsukermann. Flexible local approximation method for electro- and magnetostatics. In *IEEE Trans. Magn.*, to appear in 2004.

[97] D. van der Spoel, A. R. van Buuren, E. Apol, P. J. Meulenhoff, D. P. Tieleman, A. L. T. M. Sijbers, B. Hess, K. A. Feenstra, E. Lindahl, R. van Drunen, and H. J. C. Berendsen. *Gromacs User Manual version 3.1.* Nijenborgh 4, 9747 AG Groningen, The Netherlands. Internet: http://www.gromacs.org, 2001.

[98] W. F. van Gunsteren. Gromos96 homepage, 1996. http://www.igc.ethz.ch/gromos/.

[99] J. D. Weeks, D. Chandler, and H. C. Andersen. Role of repulsive forces in determining the equilibrium structure of simple liquids. *J. Chem. Phys.*, 54:5237, 1971.

[100] A. H. Widmann and D. B. Adolf. A comparison of ewald summation techniques for planar surfaces. *Comp. Phys. Comm.*, 107:167–186, 1997.

[101] J. Widom and R. L. Baldwin. Cation-induced toroidal condensation of dna. *J. Mol. Biol.*, 144:431–453, 1980.

[102] I.-C. Yeh and M. L. Berkowitz. Ewald summation for systems with slab geometry. *J. Chem. Phys.*, 111(7):3155–3162, 1999.

Index