

# Non-Uniform FFT and Its Applications in Particle Simulations

Yong-Lei Wang<sup>1</sup>, Fredrik Hedman<sup>2</sup>, Massimiliano Porcu<sup>1,3</sup>, Francesca Mocci<sup>1,3</sup>, Aatto Laaksonen<sup>1</sup>

<sup>1</sup>Department of Materials and Environmental Chemistry, Arrhenius Laboratory, Stockholm University, Stockholm, Sweden

<sup>2</sup>Noruna AB, Stockholm, Sweden

<sup>3</sup>Dipartimento di Scienze Chimiche e Geologiche, Cagliari University, Cittadella Universitaria di Monserrato (CA), Monserrato, Italy  
Email: [yonglei.wang@mmk.su.se](mailto:yonglei.wang@mmk.su.se), [fredrik.hedman@noruna.se](mailto:fredrik.hedman@noruna.se), [massi@dsf.unica.it](mailto:massi@dsf.unica.it), [fmocci@unica.it](mailto:fmocci@unica.it), [aatto.laaksonen@mmk.su.se](mailto:aatto.laaksonen@mmk.su.se)

Received October 19, 2013; revised November 19, 2013; accepted November 26, 2013

Copyright © 2014 Yong-Lei Wang *et al.* This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. In accordance of the Creative Commons Attribution License all Copyrights © 2014 are reserved for SCIRP and the owner of the intellectual property Yong-Lei Wang *et al.* All Copyright © 2014 are guarded by law and by SCIRP as a guardian.

## ABSTRACT

**Ewald summation method, based on Non-Uniform FFTs (ENUF) to compute the electrostatic interactions and forces, is implemented in two different particle simulation schemes to model molecular and soft matter, in classical all-atom Molecular Dynamics and in Dissipative Particle Dynamics for coarse-grained particles. The method combines the traditional Ewald method with a non-uniform fast Fourier transform library (NFFT), making it highly efficient. It scales linearly with the number of particles as  $\mathcal{O}(N \log N)$ , while being both robust and accurate. It conserves both energy and the momentum to float point accuracy. As demonstrated here, it is straightforward to implement the method in existing computer simulation codes to treat the electrostatic interactions either between point-charges or charge distributions. It should be an attractive alternative to mesh-based Ewald methods.**

## KEYWORDS

**Ewald Summation Method; Non-Uniform Fast Fourier Transform; Dissipative Particle Dynamics**

## 1. Introduction

Computer modeling and simulation of molecular systems is an established discipline rapidly expanding and widely used from Materials Science to Biological systems. At finite temperatures and normal pressures, molecular compounds are in continuous rapid motion. In liquids and solutions, molecules collide and interact with each other at close distances. They also interact with external fields. In computer simulations of molecular systems, mechanistic ball and spring models are commonly employed to give a simple picture of atoms with specific sizes and masses bonded together with covalent bonds. Molecular equilibrium geometries and interactions are defined by so-called force fields where a somewhat arbitrary division between intramolecular and intermolecular interactions is made, because this partitioning is a simplification of the more detailed and fundamental understanding furnished by a quantum mechanical description.

Intramolecular interactions are normally described by bond stretching, angle bending and torsional angle motion terms. These interactions involve the closest bonded atoms described by two-, three-, and four-body terms, respectively. The bond and angle terms are normally given as harmonic wells while the torsion term is most often expressed as a Fourier sum.

Intermolecular interactions are those between separate molecules but also include all interactions within the same molecule beyond the bonded interactions. Non-bonded interactions are further divided between short-ranged and long-ranged interactions. The short-range interactions mimic the Van der Waals type of forces, while the long-range interactions are electrostatic interactions. The electron distributions around atoms are approximated by fixed point charges, and their interactions are treated by using Coulomb's law.

The effective range of short-range interactions is limited to a specified cut-off. By assuming a uniform beyond the cut-off, correction terms to the energy can be obtained by integrating over the non-zero part of the interactions [1-3]. Thus the fast-decaying short-range interactions can be accurately approximated by truncation at the cut-off distance in most calculations.

Artificially collecting and dividing the diffuse and fluctuating electron densities inside and around molecules on single atomic sites are a crude but conceptually simple and effective approximation, since Coulomb's law can be invoked. However, this simplification comes at a price since the interactions between point charges stretch over very long distances. Furthermore, these long-range interactions cannot be truncated without introducing simulation artifacts [4-7].

As the system size grows, calculating the electrostatic interactions becomes the major computational bottleneck. Methods based on Ewald summation [8,9] are still considered as the most reliable choice, and a large variety of schemes to compute them in computer simulations have been proposed [10-16]. There is also a multitude of alternative methods for representing electrostatic interactions. Examples are methods based on a cut-off [17-19], tree and multipole based methods [20-25], multigrid methods [26-28], reaction field methods [29-31], the particle mesh method [32,33] and the isotropic sum method [34].

In this paper, we describe an approach to Ewald summation based on the non-uniform fast Fourier transform technique. We use the acronym ENUF-Ewald summation using Non-Uniform fast Fourier transform (NFFT) technique. Our method combines the traditional Ewald summation technique with the NFFT to calculate electrostatic energies and forces in molecular computer simulations. In the paper, we show that ENUF is an easy-to-implement, practical, and efficient method for calculating electrostatic interactions. Energy and momentum are both conserved to float point accuracy. By a suitable choice of parameters, ENUF can be made to behave as traditional Ewald summation but at the same time gives a computational complexity of  $\mathcal{O}(N \log N)$ , where  $N$  is the number of electrostatic interaction sites in the system. Weighing all these properties together, we believe that ENUF should be an attractive alternative in simulations where the high accuracy of Ewald summation is desired.

In the next section, we summarize the basic methodology to apply Ewald summation to computing the electrostatic energies and forces within periodic boundary conditions. In Section 3, we introduce the Ewald method where the reciprocal space part is calculated based on non-uniform FFTs and discuss the underlying concepts and use of the libraries. We also give general guidelines to implement the method in existing simulation programs. In Section 4, we discuss its implementation in a general purpose atomistic Molecular Dynamics simulation package M.DynaMix giving its scaling characteristics in a standard desktop computer. In Section 5, we demonstrate its implementation in Dissipative Particle Dynamics package. This method is applied to simulating soft charged mesoscopic particles. It is necessary to use charge distributions in order to avoid non-physical aggregation of soft charged particles if point charges are used. This issue is discussed further in Section 5.

## 2. Electrostatic Interactions

We start by describing a model system of charged particles which captures the most salient features of electrostatic interactions in general MD systems. The electrostatic potential of a system with periodic boundary conditions (PBC) is first stated; we follow with the manipulation of the basic formulas to the form in which they are commonly written; this Section ends with a summary of expressions for both energy and forces.

### 2.1. Ewald Summation

Consider a cubic simulation box with edge length  $L$ , containing  $N$  charged particles, each with a charge  $q_i$ , located at  $\mathbf{r}_i$ . The boundary conditions in a system without cut-off is represented by replicating the simulation box in all directions. The total electrostatic potential energy of the charge-charge interactions is then given by

$$U_{qq} = \sum_{i=1}^N \sum_{j=i+1}^N \frac{q_i q_j}{|\mathbf{r}_{ij}|} + \frac{1}{2} \sum_{\mathbf{n} \neq 0} \sum_{i=1}^N \sum_{j=1}^N \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} = \frac{1}{2} \sum_{\mathbf{n}} \sum_{i=1}^N \sum_{j=1}^N \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|}, \quad (1)$$

where  $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ , and  $|\mathbf{r}|$  denotes the length (2-norm) of the vector  $\mathbf{r}$ . Because of the long-range nature of the electrostatic interactions,  $U_{qq}$  includes contributions from all replicas, but exclude self-interactions, which is expressed in the triple sum in Equation (1). The outer sum is taken over all integer vectors

$\mathbf{n} = (n_1, n_2, n_3)$  such that each  $n_i \in \mathbb{Z}$ . The  $\dagger$  symbol on the first summation sign in Equation (1) indicates that the self-interaction terms should not be included, *i.e.*, when  $\mathbf{n} = 0$  then the  $i = j$  terms are omitted.

The sum in Equation (1) is not an absolutely convergent series, but rather conditionally convergent. As a consequence, the order of summation affects the value of the series. In fact, it was discovered by Riemann that any conditionally convergent series of real terms can be rearranged to yield a series which converges to any prescribed sum [35]. In a sense, this is a situation very similar to the case when a linear equation has an infinite number of solutions because it is under-determined; by adding a set of conditions a unique solution may be defined. For the specific case of Equation (1), a physically relevant summation order has to be prescribed and the boundary conditions of the surrounding media have to be specified.

The lattice sum of Equation (1) can be calculated by a method that was first developed in 1921 by Ewald [8]. He used it to calculate lattice potentials in solids. In the context of Molecular Dynamics, there are several different derivations of the Ewald summation method that are more recent; a small selection is given by [3,9]. In the following discussion we mainly follow the work of de Leeuw *et al.* [9,36,37].

In [9], de Leeuw *et al.* developed a technique using convergence factors that transforms the sum of a conditionally convergent series into a series with a well-defined sum. Furthermore, they showed that applying a specific convergence factor is equivalent to a certain summation order. Assuming an overall charge neutral system,  $\sum_i q_i = 0$ , and summing the terms in Equation (1) over all integer vectors  $\mathbf{n}$  in concentric spherical order, they showed that the electrostatic potential energy can be written as

$$U_{qq}(\epsilon_r) = \sum_{1 \leq i < j \leq N} \frac{q_i q_j}{L} \Psi\left(\frac{r_{ij}}{L}\right) + \frac{\xi}{2L} \sum_{i=1}^N q_i^2 + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left(\sum_i q_i \mathbf{r}_i\right)^2, \quad (2)$$

when the surrounding media of the periodically replicated cell is a uniform dielectric with dielectric constant  $\epsilon_r$  and distances are calculated with the minimum image convention. When the surrounding media is a conductor ( $\epsilon_r = \infty$ ), the energy can be written as

$$U_{qq}(\epsilon_r = \infty) = \sum_{1 \leq i < j \leq N} \frac{q_i q_j}{L} \Psi\left(\frac{r_{ij}}{L}\right) + \frac{\xi}{2L} \sum_{i=1}^N q_i^2 = U_{qq}(\epsilon_r) - \frac{2\pi}{(2\epsilon_r + 1)L^3} \left(\sum_i q_i \mathbf{r}_i\right)^2. \quad (3)$$

From Equation (3) it is clear that the boundary conditions, vacuum or conductor, have an effect on the energy of the system. Depending on the simulated system and the properties of interest, the choice of boundary conditions can affect the obtained results [38,39].

The function  $\Psi$  is given by

$$\Psi(\mathbf{r}) = \sum_n \frac{\text{erfc}(\alpha|\mathbf{r} + \mathbf{n}|)}{|\mathbf{r} + \mathbf{n}|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{\exp\left(2\pi i \mathbf{n} \cdot \mathbf{r} - \pi^2 |\mathbf{n}|^2 / \alpha^2\right)}{|\mathbf{n}|^2}, \quad (4)$$

with the two error functions defined as  $\text{erfc}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$  and  $\text{erf}(x) = 1 - \text{erfc}(x)$ . The number  $\xi$  used in Equations (2) and (3) is defined as

$$\xi = \sum_{n \neq 0} \frac{\text{erfc}(\alpha|\mathbf{n}|)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / \alpha^2}}{|\mathbf{n}|^2} - \frac{2\alpha}{\sqrt{\pi}}, \quad (5)$$

where  $\alpha > 0$  is a free parameter.

Equations (2) and (3) are not in a form that is appropriate for efficient numerical calculations and in the case of Molecular Dynamics simulation we also need expressions for the forces. To arrive at a more suitable form we make the necessary analysis for the electrostatic energy and forces in the following Sections.

## 2.2. Energies in Ewald Summation

To rearrange and expand Equation (2) we first insert  $r_{ij}/L$  in Equation (4)

$$\begin{aligned} \Psi\left(\frac{\mathbf{r}_{ij}}{L}\right) &= \sum_n \frac{\operatorname{erfc}\left(\alpha\left|\frac{\mathbf{r}_{ij}}{L} + \mathbf{n}\right|\right)}{\left|\frac{\mathbf{r}_{ij}}{L} + \mathbf{n}\right|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{\exp\left(2\pi i \mathbf{n} \cdot \frac{\mathbf{r}_{ij}}{L} - \pi^2 |\mathbf{n}|^2 / \alpha^2\right)}{|\mathbf{n}|^2} \\ &= L \sum_n \frac{\operatorname{erfc}\left(\frac{\alpha}{L} |\mathbf{r}_{ij} + \mathbf{n}L|\right)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / \alpha^2\right)}{|\mathbf{n}|^2}. \end{aligned} \tag{6}$$

Next we rescale  $\alpha$  by making the substitution  $\alpha \rightarrow \alpha L$  in Equations (5) and (6) to get

$$\Psi\left(\frac{\mathbf{r}_{ij}}{L}\right) = L \sum_n \frac{\operatorname{erfc}\left(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|\right)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / (\alpha L)^2\right)}{|\mathbf{n}|^2} \tag{7}$$

and

$$\xi = \sum_{n \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{n}|L)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} - \frac{2\alpha L}{\sqrt{\pi}}. \tag{8}$$

Inserting Equations (7) and (8) into Equation (2) we get

$$\begin{aligned} U_{qq}(\epsilon_r) &= \sum_{1 \leq i < j \leq N} q_i q_j \left\{ \sum_n \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi L} \sum_{n \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / (\alpha L)^2\right)}{|\mathbf{n}|^2} \right\} \\ &+ \frac{1}{2L} \sum_i q_i^2 \left[ \sum_{n \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{n}|L)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} - \frac{2\alpha L}{\sqrt{\pi}} \right] + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2. \end{aligned} \tag{9}$$

Note that the summation in Equation (9) is for  $i < j$  in the first sum. We make further simplifications by studying the terms on the right hand side of Equation (9) for  $n = 0$  and  $n \neq 0$ .

When  $n = 0$  we have the following terms

$$\begin{aligned} &\sum_{1 \leq i < j \leq N} q_i q_j \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij}|)}{|\mathbf{r}_{ij}|} - \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2 + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2 \\ &= \frac{1}{2} \sum_{i,j}^\dagger \frac{q_i q_j}{|\mathbf{r}_{ij}|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij}|) - \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2 + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2 \end{aligned} \tag{10}$$

with the terms independent of  $\mathbf{n}$  included. The  $\dagger$  symbol indicates that the  $i = j$  terms are excluded from the daggered sum.

When  $n \neq 0$  we get

$$\begin{aligned} &\sum_{1 \leq i < j \leq N} q_i q_j \left\{ \sum_{n \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi L} \sum_{n \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / (\alpha L)^2\right)}{|\mathbf{n}|^2} \right\} \\ &+ \frac{1}{2L} \sum_i q_i^2 \left\{ \sum_{n \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{n}|L)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \right\} \\ &= \frac{1}{2} \sum_{i,j} q_i q_j \sum_{n \neq 0} \left\{ \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi L} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) \right\}. \end{aligned} \tag{11}$$

The factor  $1/2$  in Equation (11) comes from changing the summation from  $i < j$  to all pairs  $i$  and  $j$ , and using the symmetry induced by  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$  and  $\pm \mathbf{n}$ .

By combining Equations (10) and (11) we identify the real-space term  $U_{qq}^{real}$ , the reciprocal-space term  $U_{qq}^{recip}$ , the self-interaction term  $U_{qq}^{self}$ , and the boundary-condition term  $U_{qq}^{bc}$ . The real-space term is given by

$$\begin{aligned} U_{qq}^{real} &= \sum_{1 \leq i < j \leq N} q_i q_j \frac{\text{erfc}(\alpha |\mathbf{r}_{ij}|)}{|\mathbf{r}_{ij}|} + \sum_{i < j} \sum_{\mathbf{n} \neq 0} \frac{\text{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{2} \sum_i q_i^2 \sum_{\mathbf{n} \neq 0} \frac{\text{erfc}(\alpha |\mathbf{n}L|)}{|\mathbf{n}L|} \\ &= \frac{1}{2} \sum_{\mathbf{n}} \sum_{i, j} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \text{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|), \end{aligned} \quad (12)$$

and the reciprocal-space term is

$$U_{qq}^{recip} = \frac{1}{\pi L} \sum_{i < j} \sum_{\mathbf{n} \neq 0} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) + \frac{1}{2\pi L} \sum_i \sum_{\mathbf{n} \neq 0} q_i q_i \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2}. \quad (13)$$

However, with the symmetries generated by  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$  and  $\pm \mathbf{n}$ , we get

$$\begin{aligned} U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{i \neq j} \sum_{\mathbf{n} \neq 0} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) + \frac{1}{2\pi L} \sum_i \sum_{\mathbf{n} \neq 0} q_i q_i \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \sum_{i, j} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right). \end{aligned} \quad (14)$$

Furthermore, we have

$$U_{qq}^{self} = \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2, \quad (15)$$

$$U_{qq}^{bc} = \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2, \quad (16)$$

and finally

$$U_{qq}(\epsilon_r) = U_{qq}^{real} + U_{qq}^{recip} - U_{qq}^{self} + U_{qq}^{bc}. \quad (17)$$

The reciprocal-space part, Equation (14), can be expanded in two different forms. The first form is in terms of the structure factor  $S(\mathbf{n})$ ,

$$S(\mathbf{n}) = \sum_i q_i \exp\left(-\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right), \quad (18)$$

and is given by

$$\begin{aligned} U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \sum_{i, j} q_i q_j \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot (\mathbf{r}_i - \mathbf{r}_j)\right) \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \sum_i q_i \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \sum_j q_j \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot (-\mathbf{r}_j)\right) \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(-\mathbf{n}) S(\mathbf{n}). \end{aligned} \quad (19)$$

Now for a fixed  $\mathbf{n}$ , the structure factor  $S(\mathbf{n})$  is just a complex number  $a$ , and the simple fact that  $a\bar{a} = \text{Re}(a)^2 + \text{Im}(a)^2$ , gives the real form of Equation (14)

$$\begin{aligned}
 U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{n \neq 0} \frac{e^{-\pi^2 |n|^2 / (\alpha L)^2}}{|n|^2} \sum_{i,j} q_i q_j \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot (\mathbf{r}_i - \mathbf{r}_j)\right) \\
 &= \frac{1}{2\pi L} \sum_{n \neq 0} \frac{e^{-\pi^2 |n|^2 / (\alpha L)^2}}{|n|^2} \left[ \left( \sum_i q_i \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 + \left( \sum_i q_i \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 \right].
 \end{aligned}
 \tag{20}$$

The first form in Equation (19) is used in our fast approach to calculating the reciprocal-space part. The last form in Equation (20) is the most common point-of-departure when implementing the reciprocal-space part.

### 2.3. Forces in Ewald Summation

Now that we have calculated the electrostatic energy of the system we can easily compute the electrostatic forces  $F_i$  that act on each particle  $i$ . Splitting the forces in the same way as we have split the energy and using Equation (17) we get the total electrostatic force by finding the negative of the gradient of the electrostatic energy

$$F_i = -\nabla_i U_{qq} = -\nabla_i \left[ U_{qq}^{real} + U_{qq}^{recip} - U_{qq}^{self} + U_{qq}^{bc} \right] = F_i^{real} + F_i^{recip} - 0 + F_i^{bc}. \tag{21}$$

The subscript  $i$  on the  $\nabla$  operator indicates that we take the partial derivatives with respect to the position of particle  $\mathbf{r}_i$  and the 0 in Equation (21) comes from the self-interaction term in Equation (15) being independent of  $\mathbf{r}_i$ . Before we do this calculation we note a couple basic, but helpful, formulas for calculating derivatives

$$\begin{aligned}
 \nabla_i \mathbf{n} \cdot \mathbf{r}_{ij} &= \nabla_i \mathbf{n} \cdot (\mathbf{r}_i - \mathbf{r}_j) = \mathbf{n} \\
 \nabla_i \frac{1}{|\mathbf{r}_{ij}|} &= -\frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^3} \\
 \nabla_i |\mathbf{r}_{ij}| &= \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \\
 \nabla_i \operatorname{erfc}(\alpha |\mathbf{r}_{ij}|) &= -\frac{2\alpha}{\sqrt{\pi}} \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \exp\left(-\alpha^2 |\mathbf{r}_{ij}|^2\right) \\
 \nabla_i \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ji}\right) &= -\mathbf{n} \frac{2\pi i}{L} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ji}\right) \\
 \nabla_i \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) &= +\mathbf{n} \frac{2\pi i}{L} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right)
 \end{aligned}
 \tag{22}$$

With above formulas, it is straightforward to find the different terms of  $F_i$ . The contribution from the real-space term,  $F_i^{real}$ , becomes

$$\begin{aligned}
 F_i^{real} &= -\nabla_i U_{qq}^{real} = -\nabla_i \left[ \frac{1}{2} \sum_n^\dagger \sum_{i,j} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|) \right] \\
 &= \sum_n^\dagger \sum_j q_i q_j \frac{\mathbf{r}_{ij} + \mathbf{n}L}{|\mathbf{r}_{ij} + \mathbf{n}L|^2} \left[ \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{2\alpha}{\sqrt{\pi}} \exp\left(-\alpha^2 |\mathbf{r}_{ij} + \mathbf{n}L|^2\right) \right],
 \end{aligned}
 \tag{23}$$

such that when  $n \neq 0$  include all  $j$  and otherwise only  $j \neq i$  - "the daggered saviour". Equation (20) is convenient to use when calculating the reciprocal-space contribution because it is expressed in terms of charge locations  $\mathbf{r}_i$  rather than relative distances  $\mathbf{r}_{ij}$ . Thus the reciprocal-space force is given by

$$\begin{aligned}
F_i^{recip} &= -\nabla_i U_{qq}^{recip} \\
&= -\nabla_i \left[ \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \left( \sum_i q_i \cos \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \right)^2 + \left( \sum_i q_i \sin \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \right)^2 \right] \right] \\
&= \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \sin \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \operatorname{Re}(S(\mathbf{n})) + \cos \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \operatorname{Im}(S(\mathbf{n})) \right].
\end{aligned} \tag{24}$$

Finally, the contribution that depends on the boundary condition

$$F_i^{bc} = -\nabla_i \left[ \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2 \right] = -\frac{4\pi}{(2\epsilon_r + 1)L^3} q_i \sum_j q_j \mathbf{r}_j. \tag{25}$$

#### 2.4. Formulas for Energy and Forces in Ewald Summation

Consider a periodically replicated system, with the central box consisting of  $N$  point charges  $q_i$  and  $\sum_i q_i = 0$ . Assume that the surrounding media at the boundary of the periodically replicated system is a uniform dielectric with dielectric constant  $\epsilon_r$ . The cubic box has edge length  $L$ ; each charge  $q_i$  is located at  $\mathbf{r}_i$ , and distances are calculated with the minimum image convention. After expansion and rearrangements of Equation (2), rescaling  $\alpha \rightarrow \alpha L$  and using symmetries induced by  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$  and  $\pm \mathbf{n}$ , the total electrostatic energy of the system can be written as

$$U_{qq}(\epsilon_r) = U_{qq}^{real} + U_{qq}^{recip} - U_{qq}^{self} + U_{qq}^{bc} \tag{26}$$

with the different terms given by

$$U_{qq}^{real} = \frac{1}{2} \sum_{\mathbf{n}} \sum_{i,j} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|), \tag{27}$$

$$\begin{aligned}
U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \sum_{i,j} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp \left( \frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} \right) \\
&= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \left( \sum_i q_i \cos \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \right)^2 + \left( \sum_i q_i \sin \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \right)^2 \right] \\
&= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) S(-\mathbf{n}),
\end{aligned} \tag{28}$$

$$U_{qq}^{self} = \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2, \tag{29}$$

$$U_{qq}^{bc} = \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2. \tag{30}$$

Note that  $\alpha > 0$  is a free parameter. The structure factor  $S(\mathbf{n})$  is defined as

$$S(\mathbf{n}) = \sum_i q_i \exp \left( -\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i \right). \tag{31}$$

The total electrostatic force,  $F_i$ , on each particle is

$$F_i = F_i^{real} + F_i^{recip} + F_i^{bc}. \tag{32}$$

Each of the force terms given by

$$F_i^{real} = \sum_{\mathbf{n}} \sum_j q_i q_j \frac{\mathbf{r}_{ij} + \mathbf{n}L}{|\mathbf{r}_{ij} + \mathbf{n}L|^2} \left[ \frac{\text{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{2\alpha}{\sqrt{\pi}} \exp(-\alpha^2 |\mathbf{r}_{ij} + \mathbf{n}L|^2) \right], \quad (33)$$

$$F_i^{recip} = \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \sum_j q_j \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_j\right) - \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \sum_j q_j \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_j\right) \right], \quad (34)$$

$$F_i^{bf} = -\frac{4\pi}{(2\epsilon_r + 1)L^3} q_i \sum_j q_j \mathbf{r}_j. \quad (35)$$

The positive number  $\alpha$ , the so called Ewald convergence parameter, is chosen for computational convenience. Observe that  $\text{erfc}(x) \approx e^{-x^2}$  for large values of  $x$ . By choosing  $\alpha$  large enough in Equation (27), we can ensure that the only terms that contribute in the real-space sum is when  $\mathbf{n} = 0$ . This may be expressed so that all terms with  $|\mathbf{n}| < n_{cut}$  should be included.

Choose a cut-off in both real-space and reciprocal-space so that the neglected terms in the real-space and reciprocal-space parts are of the same order  $\delta$ , or less. The truncation in real-space implies that a sufficient number of terms must be included in the reciprocal-space sums, Equation (28).

Given a required accuracy  $\delta$ ,  $n_{cut}$  is fixed by

$$e^{-\pi^2 n_{cut}^2 / (\alpha L)^2} \leq \delta \Rightarrow n_{cut} \geq \frac{\alpha L}{\pi} \sqrt{-\log(\delta)}, \quad (36)$$

and  $r_{cut}$  is determined by

$$\text{erfc}(\alpha r_{cut}) \approx e^{-\alpha^2 r_{cut}^2} \leq \delta \Rightarrow r_{cut} = \frac{\pi n_{cut}}{\alpha^2 L}. \quad (37)$$

We have two conditions and four parameters. With a required  $\delta$  we may just as well pick a suitable value for  $n_{cut}$  and let the above two equations determine  $\alpha$  and  $r_{cut}$ .

With an optimal choice of parameters the computational effort of the Ewald method becomes  $\mathcal{O}(N^{3/2})$  [36] giving a considerable improvement over the  $\mathcal{O}(N^2)$  computational complexity implied by the ‘infinite’ reach of the Coulomb interactions.

### 3. ENUF: A Fast Method for Calculating Electrostatic Interactions

In Section 2 we summarized known results and prepared the ground for the development of a fast method for Ewald summation using the discrete nonuniform fast Fourier transform (NDFT).

#### 3.1. Discrete Fourier Transforms for Non-Equispaced Data

The fast Fourier transform for nonuniform data-points (NFFT) [40] is a generalization of the FFT [41]. Several similar approaches have been proposed; some examples are [42-50] with comparisons in [47,51,52].

The basic idea of NFFT is to combine the standard FFT and linear combinations of a window function that is well localized in both the spatial domain and the frequency domain. A controlled approximation using a cut-off in the frequency domain and a limited number of terms in the spatial domain results in an aliasing error and a truncation error, respectively. The aliasing errors is controlled by the oversampling factor  $\sigma_s$ , and the truncation error is controlled by the number of terms,  $m$ , in the spatial/time approximation. For a number of window functions (Gaussian, B-spline, Sinc-power, Kaiser-Bessel), it has been shown that for a fixed oversampling factor,  $\sigma_s > 1$ , the error decays exponentially with  $m$  [53].

##### 3.1.1. Problem Definition

We wish to calculate the discrete Fourier transform for nonequispaced data. The problem can be stated as follows. For a finite number of given Fourier coefficients  $\hat{f}_k \in \mathcal{C}$  with  $k \in I_M$  we want to evaluate the trigonometric polynomial  $f(x) = \sum_{k \in I_M} \hat{f}_k \exp(-2\pi i k x)$  at each of the given nonequispaced points  $(\mathbf{x}_j \in \mathcal{D}^d, j = 0, \dots, N - 1)$ . In the literature, *points* are often called *knots*. We use the two terms synonymously.



Obviously, the details of an NDFT depend on the definitions of a sampling set for knots,  $\mathcal{D}^d$ , and an index space  $I_M$ . More in-depth discussions and further details can be found in [53,54]. The presentation that follows is mainly drawn from these sources.

3.1.2. Underlying Concepts

Consider a  $d$ -dimensional domain  $\mathcal{D}^d$  in which the set of nonequispaced knots, or data points, are located. Let

$$\mathcal{D}^d := \left\{ \mathbf{x} = (x_t)_{t=0, \dots, d-1} \in \mathcal{R}^d : -\frac{1}{2} \leq x_t < \frac{1}{2}, t = 0, \dots, d-1 \right\}, \tag{38}$$

and the set of  $N$  data points  $(\mathcal{X} := \{ \mathbf{x}_j \in \mathcal{D}^d : j = 0, \dots, N-1 \})$ . For the application we have in mind  $d$  is usually 2 or 3. Let  $\mathcal{F}$  be a function space of trigonometric polynomials with degree  $M_t (t = 0, \dots, d-1)$  in dimension  $t$ ; the function space  $\mathcal{F}$  can be defined as

$$\mathcal{F} := \left\{ f : \mathcal{D}^d \mapsto \mathcal{C} \text{ such that } f \in \text{span} \left( e^{-2\pi i \mathbf{k} \cdot \mathbf{x}} : \mathbf{k} \in I_M \right) \right\}. \tag{39}$$

The dimension of this function space is  $\dim(\mathcal{F}) = M_{\Pi}$ , where  $M_{\Pi} = \prod_{t=0}^{d-1} M_t$ . The frequencies  $\mathbf{k} \in I_M$  with the index set  $I_M$  are such that

$$I_M := \left\{ \mathbf{k} = (k_t)_{t=0, \dots, d-1} \in \mathcal{Z}^d : -\frac{M_t}{2} \leq k_t < \frac{M_t}{2}, t = 0, \dots, d-1 \right\}. \tag{40}$$

3.1.3. Matrix-Vector Formulation

With these preliminary definitions we carry on with the problem of calculating the discrete Fourier transform for nonequispaced data. For a finite number of given Fourier coefficients  $\hat{f}_{\mathbf{k}} \in \mathcal{C}$  with  $\mathbf{k} \in I_M$  we want to evaluate the trigonometric polynomial  $f(\mathbf{x}) = \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x})$  at each of the given nonequispaced knots in  $\mathcal{X}$ , where the product  $\mathbf{k} \mathbf{x}$  is the usual scalar product of the two vectors  $\mathbf{k}$  and  $\mathbf{x}$  as  $\mathbf{k} \mathbf{x} := k_0 x_0 + \dots + k_{d-1} x_{d-1}$ . Consequently, for each  $\mathbf{x}_j \in \mathcal{X}$ , we evaluate  $f_j = f(\mathbf{x}_j) := \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x}_j)$ . This may be reformulated in matrix-vector notation by setting

$$\begin{aligned} f &:= (f_j)_{j=0, \dots, N-1}, \\ \mathbf{A} &:= \left( e^{-2\pi i \mathbf{k} \mathbf{x}_j} \right)_{j=0, \dots, N-1; \mathbf{k} \in I_M}, \\ \hat{\mathbf{f}} &:= (\hat{f}_{\mathbf{k}})_{\mathbf{k} \in I_M} \end{aligned} \tag{41}$$

and writing  $f = \mathbf{A} \hat{\mathbf{f}}$ .

3.1.4. Related Matrix-Vector Products

A number of related NDFT matrix-vector products can also be defined. To write them down we let  $\bar{A}$  be the complex conjugate of the elements of the matrix  $A$  and  $A^H = \bar{A}^T$  the transposed complex conjugate of the matrix  $A$ . Using these conventions we can name and summarize the related NDFT matrix-vector products and their component representation as

$$\text{Regular : } f = \mathbf{A} \hat{\mathbf{f}}, f_j = \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x}_j). \tag{42}$$

$$\text{Adjoint : } \hat{\mathbf{f}} = \mathbf{A}^H f, \hat{f}_{\mathbf{k}} = \sum_{j=0}^{N-1} f_j \exp(2\pi i \mathbf{k} \mathbf{x}_j). \tag{43}$$

$$\text{Conjugated : } f = \bar{\mathbf{A}} \hat{\mathbf{f}}, f_j = \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(2\pi i \mathbf{k} \mathbf{x}_j). \tag{44}$$

$$\text{Transposed : } \hat{\mathbf{f}} = \mathbf{A}^T f, \hat{f}_{\mathbf{k}} = \sum_{j=0}^{N-1} f_j \exp(-2\pi i \mathbf{k} \mathbf{x}_j). \tag{45}$$

### 3.1.5. NDFT, FFT and NFFT

From the different NDFT products written in matrix-vector form, as in Equations (42)-(45), it is clear that it takes  $\mathcal{O}(NM_{\Pi})$  arithmetic operations to transform between the Fourier-samples and the Fourier-coefficients.

This is simply because the matrix  $A$  is  $N \times M_{\Pi}$ , with  $M_{\Pi} = \dim(\mathcal{F}) = \prod_{t=0}^{d-1} M_t$ .

However, for the special case of  $M_t = M_{(t=0, \dots, d-1)}$  and  $N = M^d$  equispaced knots  $\mathbf{x}_k$  ( $k \in I_M$ ), the Fourier-samples  $f_k$  can be calculated from the Fourier-coefficients  $\hat{f}_k$  by the fast Fourier transform (FFT) with  $\mathcal{O}(N \log N)$  arithmetic operations.

The fast Fourier transform for nonequispaced knots (NFFT) is a generalization of the FFT. The essential idea is that of combining a window function with the standard FFT. The window function is a well localized function in both the space domain and frequency domain. Several different window functions and similar approaches have been proposed. The resulting algorithms are approximate and some of them have been shown to have a computational complexity of  $\mathcal{O}(M_{\Pi} \log M_{\Pi} + \log(1/\epsilon)N)$ , where  $\epsilon$  is the desired accuracy [53].

## 3.2. Fast Ewald Summation

Using optimal parameters in the Ewald summation method implies that the time to calculate the real-space part and the reciprocal-space part are approximately equal. As the number of particles in the system grows we would like to combine the calculation of the short-range part of the potential with the real-space part. This implies that we need to choose a real-space cut-off about the same size as the short-range cut-off. With this nonoptimal choice, the reciprocal-space parts of the Ewald summation method become the most time-consuming to calculate [55].

To show how a fast Ewald summation approach may be obtained from the regular Ewald method, described in Section 2, we focus on the reciprocal-space parts. In Section 3.1, we gave the details of the discrete Fourier transform (DFT) for data that is nonuniformly spaced (NDFT). Based on these definitions we get a number of useful algorithmic primitives. First we reformulate the reciprocal-space part of the regular Ewald method in terms of the NDFT primitives. Then we show how the fast Fourier transform for nonequispaced (NFFT) can be applied, yielding an Ewald method based on the nonuniform fast Fourier transform.

### 3.2.1. Reciprocal Space Terms as DFT

We apply the generalized DFT, described in Section 3.1, to the calculation of the reciprocal-space energy and forces. This allows us to formulate the standard Ewald method for calculating the reciprocal energy and forces in terms of the NDFT primitives.

### 3.2.2. Reciprocal Energy

In the case of the electrostatic energy we have from Equation (28)

$$U_{qq}^{recip} = \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) S(-\mathbf{n}), \quad (46)$$

with the structure factor  $S(\mathbf{n})$  defined as

$$S(\mathbf{n}) = \sum_j q_j \exp\left(-\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_j\right). \quad (47)$$

By comparing the definition of the transposed NDFT in Equation (45) and the structure factor in Equation (47) we note that they have the same structure; after a renumbering of the location indexes, the summation limits are also the same. In fact, by setting the normalized locations,  $\mathbf{x}_j = \mathbf{r}_j/L$ , and the samples,  $\hat{f}_j = q_j$ , we see by inspection that Equation (47) is a 3D instance of Equation (45) with  $\hat{f}_n = S(\mathbf{n})$ . Furthermore, assuming that the MD simulation box is centered around the origin, the normalized locations can be assumed to be in the domain  $\mathcal{D}^d$  as defined in Equation (38).

Consequently, we can use the NDFT approach to calculate each of the components of the structure factor. From a computational point of view this means that we can also expect to utilize an NFFT based algorithm to calculate the components of the structure factor  $S(\mathbf{n})$ , rather than the straightforward summation normally used in the Ewald method.

Recasting Equation (46) in terms of Fourier-components

$$U_{qq}^{recip} = \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) S(-\mathbf{n}) = \frac{1}{\pi L} \sum_{n_z > 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} |\hat{\mathbf{f}}_{\mathbf{n}}|^2. \quad (48)$$

Calculating the energy,  $U_{qq}^{recip}$ , using Equation (48) means that we

- 1) calculate all  $\hat{\mathbf{f}}_{\mathbf{n}}$  using the transposed NDFT,
- 2) scale each  $|\hat{\mathbf{f}}_{\mathbf{n}}|^2$  with a factor given by Equation (48),
- 3) sum all the scaled components.

### 3.2.3. Reciprocal Forces

We calculate the contribution from the reciprocal-space forces using a similar approach as for the energy. In the formula below,  $\text{Re}(\cdot)$ , and  $\text{Im}(\cdot)$ , denote the real and imaginary part of the arguments, respectively. From Equation (24) we have that

$$F_i^{recip} = \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \text{Re}(S(\mathbf{n})) + \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_j\right) \text{Im}(S(\mathbf{n})) \right]. \quad (49)$$

Now, the structure factor  $S(\mathbf{n})$  is just a complex number so the expression in the brackets in Equation (49) can be written as the imaginary part of a product

$$\sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \text{Re}(S(\mathbf{n})) + \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \text{Im}(S(\mathbf{n})) = \text{Im} \left[ \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right) S(\mathbf{n}) \right]. \quad (50)$$

Inserting Equation (50) this into Equation (49) gives

$$\begin{aligned} F_i^{recip} &= \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \text{Im} \left[ \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right) S(\mathbf{n}) \right] \\ &= \text{Im} \left[ \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right) S(\mathbf{n}) \right]. \end{aligned} \quad (51)$$

Note that Equation (51) is a vector equation. Furthermore, each of the three components has the same structure as the conjugated NDFT of Equation (44). By setting the normalized locations,  $\mathbf{x}_j = \mathbf{r}_j / L$ , and the samples,

$$\hat{\mathbf{g}}_{\mathbf{n}} = \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) \quad \text{for } \mathbf{n} \neq 0, \quad (52)$$

we see, again, by inspection that each component of Equation (51) is a 3D instance of Equation (44). Assuming that  $\mathbf{n}$  is in the index set  $I_M$  of Equation (40),  $\mathbf{n} \in I_M$ , and setting  $\mathbf{g}_0 = 0$ , we can formulate  $F_i^{recip}$  directly in Fourier-terms

$$F_i^{recip} = \text{Im} \left[ \frac{2q_i}{L^2} \sum_{\mathbf{n} \in I_M} \hat{\mathbf{g}}_{\mathbf{n}} e^{2\pi i \mathbf{n} \cdot \mathbf{x}_i} \right] = \frac{2q_i}{L^2} \text{Im}(\mathbf{g}_i). \quad (53)$$

Calculating the reciprocal-space force  $F_i^{recip}$  on particle  $i$ , using Equation (53), means that we

- 1) start with the structure factor components,  $S(\mathbf{n})$ , already obtained when we calculated  $U_{qq}^{recip}$ ,
- 2) scale each  $S(\mathbf{n})$  using Equation (52),
- 3) giving a new set of Fourier-coefficients that are transformed back to real-space, via Equation (53), using the conjugated NDFT, and finally

- 4) with Equation (53), taking the imaginary part of coefficient  $g_i$  and scaling it with  $\frac{2q_i}{L^2}$  gives the

reciprocal force on particle  $i$ .

Thus we can use the NDFT approach to calculate each of the components of the reciprocal forces. Again, from a computational point of view this means we can expect to utilize an NFFT based algorithm to find the respective components.

### 3.2.4. Combining the Ewald Method and NFFT

The reformulation of  $U_{qq}^{recip}$  and  $F_i^{recip}$  as in Equations (48) and (53) shows the central role of the transposed and conjugated NDFT in calculating the reciprocal-space energy and forces. Starting with the location of the charged particles, the structure factor  $S(\mathbf{n})$  is calculated via a transposed NDFT. In the language of Ewald summation, we transform from real-space to reciprocal-space. Scaling the absolute value of the Fourier-components and summing gives  $U_{qq}^{recip}$ . To find  $F_i^{recip}$  we go back from the reciprocal-space to the real-space by first calculating the Fourier-components of the forces and then performing a conjugated NDFT.

An implementation of Ewald summation uses cut-offs, in reciprocal-space,  $n_{cut}$ , and real-space,  $r_{cut}$ ; with  $\alpha$  large enough and with a required accuracy,  $\delta$ , truncate the sums Equation (27) and Equation (28) at the respective cut-offs so that the last term added  $\leq \delta$ , in each of the sums. When  $n_{cut}$  is fixed by

$$e^{-\pi^2|\mathbf{n}|^2/(\alpha L)^2} \leq \delta \Rightarrow n_{cut} \geq \frac{\alpha L}{\pi} \sqrt{-\log(\delta)} \Rightarrow n_{cut} \propto L \propto N^{1/3}. \quad (54)$$

Then  $r_{cut}$  is determined by

$$\text{erfc}(\alpha r_{cut}) \approx e^{-\alpha^2 r_{cut}^2} \leq \delta \Rightarrow r_{cut} = \frac{\pi n_{cut}}{\alpha^2 L}. \quad (55)$$

We have recast  $U_{qq}^{recip}$  and  $F_i^{recip}$  in terms of Fourier-components and set  $M_t = 2\sigma_s n_{cut}$ , where  $\sigma_s$  is the oversampling factor. This gives  $M_{\Pi} = \prod_{t=0}^{d-1} M_t$ . In general the computational complexity of the NFFT method is  $\mathcal{O}(M_{\Pi} \log M_{\Pi} + \log(1/\epsilon)N)$ , where  $\epsilon$  is the desired accuracy in the approximation used within NFFT [53].

Using Equation (54) and the above definition of  $M_{\Pi}$ , we see that the complexity becomes  $\mathcal{O}(N \log N + \log(1/\epsilon)N)$ . Note that  $\epsilon$  is a function of  $m$ , for a fixed oversampling factor. With a controlled approximation of the structure factor via the use of nonuniform fast Fourier transform, the original computational complexity of  $\mathcal{O}(N^{3/2})$  becomes  $\mathcal{O}(N \log N)$ .

At this stage, the path to a fast Ewald method should now be clear. By specifying an accuracy  $\delta$ , we replace the transposed and conjugated NDFT with the corresponding operations using the NFFT algorithm. Thus Equations (48) and (53) become a concise procedure to calculate approximations of  $U_{qq}^{recip}$  and  $F_i^{recip}$ . Most of the mathematical details can be kept separate and hidden in a set of library routines and the remaining formulas pertain to the physics of the problem. Furthermore, with a library implementation based on a state-of-the-art FFT-library, we have good reason to expect it to be efficient.

### 3.2.5. Implementation and Results

In a first implementation [56,57], we used the libraries FFTW [58] and NFFT [54]. Details of the accuracy and scaling properties can be found in reference papers.

Basing the implementation on libraries has a number of advantages. It makes the implementation task easier and introduces a convenient division of labor in the program code: the mathematical aspects are mainly concentrated to the libraries while the physical aspects of the problem remain. Also, since the code becomes quite compact without becoming convoluted, it becomes easier to check, understand, and explain. Improvements and optimizations of the libraries can be easily included in the program, usually by just relinking the program. For example, the customization of the window function used in the NFFT algorithm---Gaussian functions, dilated cardinal B-splines, Sinc functions, or Kaiser-Bessel functions---is currently achieved by recompiling the NFFT library and relinking the application. Due to the comparatively small size of the NFFT library this is very quick. Furthermore, improvements in either theory or implementation of the used libraries will be easily accessible.

In summary, we claim that the ENUF method is

- efficient and concise, and
- has a clear separation of concerns between mathematical and physical details.

In a sense it can be said that we get the best of two worlds: a concise and efficient algorithm. The separation

of mathematical concerns is a bonus that has the potential to simplify implementation and further developments due to the fact that they may occur independently of each-other.

#### 4. ENUF Implemented in M.DynaMix

M.DynaMix is a highly modular general purpose parallel Molecular Dynamics code for simulations of arbitrary mixtures of either rigid or flexible molecules. It was released by Lyubartsev and Laaksonen in late 90's [59]. Most common force fields can be used in simulations with a variety of periodic boundary conditions (cubic, rectangular, hexagonal or a truncated octahedron). Quantum corrections to the atomic motion can be done using the Path Integral Molecular Dynamics approach. M.DynaMix has been used in applications from materials design to biological processes.

M.DynaMix deals with particle system interacting by a force field consisting of Lennard-Jones, electrostatic, covalent bonds, angles and torsion angles potentials as well as of some optional terms, in a periodic rectangular, hexagonal or truncated octahedron cell. Rigid bonds are constrained by the SHAKE algorithm [60]. In case of flexible molecular models the double time step algorithm is used [61]. Algorithms for NVE, NVT and NPT statistical ensembles are implemented, as well as Ewald sum approach for treatment of the electrostatic interactions. An option to calculate free energy by the expanded ensemble method with Wang-Landau optimization of the balancing factor is included in later versions. For its features and capabilities, M.DynaMix is diffused in a large modeling and simulation community. Written in FORTRAN 77, it can be run on a wide variety of hardware architectures both in sequential and parallel execution. The entire program source code consists of a number of FORTRAN files (modules), made of blocks of subroutines, performing different tasks or groups of tasks.

##### 4.1. Framework Overview

The situation outlined for M.DynaMix is very common in the field of computational science: a package created in late 90's that is still at work in spite of new innovations in computing platforms. In a situation like this, each upgrade requires a trade-off between the need to preserve the existing structure and the wish to obtain as much efficiency as possible.

To take the best advantage from latest programming techniques and tools, we decide to use C language to create some of the new code segments. On one hand, modules in FORTRAN and C can easily coexist in the same code, just taking account of a few mild guidelines [62]; on the other hand, C language provides several enhancements related to intrinsic features (dynamic memory allocation and direct pointers reference) and possibility to set up complex data structures; furthermore, it allows a more plain access to a large amount of external routine libraries written in C. From a general point of view, M.DynaMix code has been projected with a good modularity degree and has been possible to make the most part of upgrades just switching a block (subroutine) with a new one.

##### 4.2. Input Parameters

To use ENUF method for treating long range interactions, M.DynaMix user needs to set the proper key string in the input file [63]; parameters to specify are the Ewald convergence factor  $\alpha$  and the number of points for FFT grid in  $x$ ,  $y$  and  $z$  direction; starting from them, the program automatically sets proper values for over-sampling factor  $\sigma_s$  and approximation parameter  $p$ .

##### 4.3. Implementation

Ewald-like methods for computing electrostatic energies typically replace part of the summation in real space with an equivalent summation in Fourier space; among M.DynaMix modules, there is one devoted to this reciprocal space duty. In the starting setup, this module was present in M.DynaMix in a version related to the full Ewald algorithm; ENUF implementation involved the creation of a second instance of this module.

A scheme of the ENUF module as implemented in M.DynaMix is presented in **Table 1**; the module in this case is a *group* of files, part of them written in FORTRAN and part in C; each file contains routines related to the algorithm steps. The table displays for each step the programming language used to write corresponding file; Some files make calls to external libraries, typically used to perform non uniform Fourier transform and its inverse.

**Table 1.** ENUF algorithm scheme: for each step is listed the language of the related routines and the presence of calls to external libs.

| Step                  | Language | Lib Calls |
|-----------------------|----------|-----------|
| 1: Coordinate scaling | F77      | No        |
| 2: <i>Adjoint</i> FFT | C        | Yes       |
| 3: Energy evaluation  | F77      | No        |
| 4: Factor rescaling   | F77      | No        |
| 5: <i>Regular</i> FFT | C        | Yes       |
| 6: Forces evaluation  | F77      | No        |

#### 4.3.1. Coordinate Scaling and Adjoint Transform

In the first step of the algorithm, particle coordinates are rescaled respect to the simulation box, in order to obtain the proper set of nonequispaced knots as required in Equation (38). Knots are stored in the unidimensional FORTRAN array  $X$ , in consecutive way, saving coordinates in reverse component order  $\{z, y, x\}$  for each knot. Particle charges are located in unidimensional array  $Q$ .

At the next step, a C routine performs the 3-dim *adjoint* transform of charges  $Q$  located at knots  $X$ . C routine access to 1-dim FORTRAN vectors  $X$  and  $Q$  as three dimensional arrays; the reverse component order adopted in  $X$  vector filling is related to this and depends on the different way to arrange multi-dimensional arrays in memory: by columns for FORTRAN, by rows for C. Transformed data are stored in the *complex* array  $\tilde{S}$ .

#### 4.3.2. Energy, Regular Transform and Forces

In this stage, energy is evaluated by a sum, according to formula (48). Then, the transformed array  $\tilde{S}$  is rescaled using Equation (51); starting from  $\tilde{S}$ , three complex rescaled arrays are created,  $\tilde{S}_x$ ,  $\tilde{S}_y$  and  $\tilde{S}_z$ , one for each direction in the reciprocal space. A C routine back transform complex data set  $\tilde{S}_x$ ,  $\tilde{S}_y$  and  $\tilde{S}_z$  related to the same set of knots  $X$ . In this step, three independent back transformations are performed, producing complex arrays  $F_x$ ,  $F_y$  and  $F_z$ . Forces contributions for each particle are obtained according to Equation (53); components in  $x$ ,  $y$  and  $z$  directions are evaluated using the imaginary part of arrays  $F_x$ ,  $F_y$  and  $F_z$ .

#### 4.4. External Libraries

To perform Fourier transform on nonequispaced data, our current implementation make use of an external existing function library. Among the available resources, we select the library NFFT [54]. NFFT is a widely diffused C subroutine library for computing nonequispaced discrete Fourier transform in one or more dimensions, of arbitrary input size and of complex data; it is based on FFTW [58].

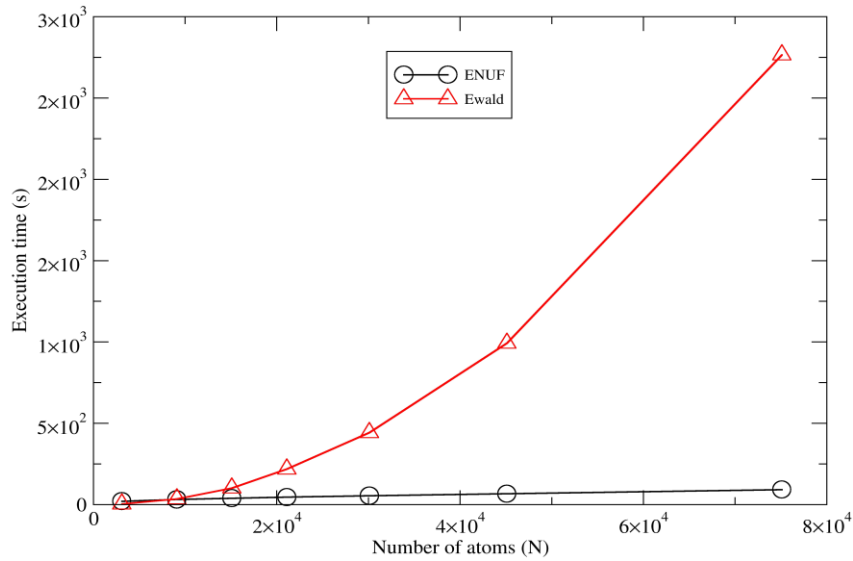
#### 4.5. Validation

In the original version of M.DynaMix the working method for the treatment of long range interactions is the full direct Ewald summation. Using a *cross-check* approach, namely comparing ENUF method to the stable one, it has been possible to debug every step of the algorithm implementation and check its correctness in deep way.

After the implementation, the same cross-check mechanism has been used to investigate ENUF precision and efficiency. **Figure 1** displays the execution time spent for long range interactions evaluation respect to the number of particles  $N$ , when full Ewald and ENUF are used. In the simulation, of 100 iterations, the system is composed by 50  $\text{Na}^+$  ions and 50  $\text{Cl}^-$  ions in water solution in a cubic cell. The number of water molecules is increased from  $10^3$  to  $25 \times 10^3$  in seven discrete steps, keeping constant the density to  $1.02 \text{ g/cm}^3$ . The simulation run on a Intel Xeon workstation, 1.86 GHz, with ram 8 Gb.

The full Ewald method has been taken as reference; for each system size, ENUF parameters have been tuned to obtain the same Ewald precision, evaluated in terms of the electrostatics energy contribution, tolerating a maximum divergence of 2%.

The plot shows that execution time for long range interactions that represents the efficiency bottleneck in the original M.DynaMix version, has been drastically reduced saving the same numerical precision. The plot also shows the different trend for execution time vs  $N$  in the two methods; the point set for the full Ewald scheme



**Figure 1.** Execution time for long range interactions vs. number of particles  $N$ , for Ewald and ENUF. System is composed by 50  $\text{Na}^+$  and 50  $\text{Cl}^-$  in  $\text{H}_2\text{O}$  solution in a periodic cubic cell. Number of water molecules is increased from  $10^3$  to  $25 \times 10^3$  keeping density to  $1.02 \text{ g/cm}^3$ .

is well fitted using the well-known theoretical behavior  $\mathcal{O}(N^2)$ ; data points related to ENUF correspond quite well to the expected  $\mathcal{O}(N \log N)$ .

### 5. Implementation of ENUF Method in Dissipative Particle Dynamics Scheme

The Dissipative Particle Dynamics (DPD) simulation method, originally proposed by Hoogerbrugge and Koelman, is a particle-based simulation method to simulate hydrodynamic phenomena at mesoscopic level [64, 65,66]. In DPD model, several atoms or molecules are grouped together to form coarse-grained particles. The interactions between any pair of DPD particles  $i$  and  $j$  are normally composed of three pairwise additive forces: the conservative force  $\mathbf{F}_{ij}^{C,DPD}$ , the dissipative force  $\mathbf{F}_{ij}^{D,DPD}$ , and the random force  $\mathbf{F}_{ij}^{R,DPD}$

$$\mathbf{f}_{ij} = \sum_{i \neq j} (\mathbf{F}_{ij}^{C,DPD} + \mathbf{F}_{ij}^{D,DPD} + \mathbf{F}_{ij}^{R,DPD}), \quad (56)$$

with

$$\mathbf{F}_{ij}^{C,DPD} = \alpha_{ij} \omega^C(r_{ij}) \hat{\mathbf{r}}_{ij}, \quad (57)$$

$$\mathbf{F}_{ij}^{D,DPD} = -\gamma \omega^D(r_{ij}) (\mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij}) \hat{\mathbf{r}}_{ij}, \quad (58)$$

$$\mathbf{F}_{ij}^{R,DPD} = \sigma \omega^R(r_{ij}) \theta_{ij} \hat{\mathbf{r}}_{ij}, \quad (59)$$

where  $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ ,  $r_{ij} = |\mathbf{r}_{ij}|$ ,  $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$ , and  $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ . The parameters  $\alpha_{ij}$ ,  $\gamma$ , and  $\sigma$  determine the strength of the conservative, dissipative, and random forces, respectively.  $\theta_{ij}$  is a randomly fluctuating variable, with zero mean and unit variance.

The pairwise conservative force is written in terms of a weight function  $\omega^C(r_{ij})$ , where  $\omega^C(r_{ij}) = 1 - r_{ij}/R_c$  is usually chosen for  $r_{ij} \leq R_c$  and  $\omega^C(r_{ij}) = 0$  for  $r_{ij} > R_c$  such that the conservative force is soft and repulsive. The unit of length  $R_c$  is related to the volume of DPD particles. Two weight functions  $\omega^D(r_{ij})$  and  $\omega^R(r_{ij})$  for dissipative and random forces, respectively, are coupled together through the fluctuation-dissipation theorem

$$\omega^D(r) = [\omega^R(r)]^2 \quad \text{and} \quad \sigma^2 = 2\gamma k_B T \quad (60)$$

to form a thermostat and generate naturally the canonical distribution (constant number of particles,  $N$ , volume,  $V$ , and temperature,  $T$ ) [67]. In most applications, the weight function  $\omega^D(r)$  adopts a simple form as [68]

$$\omega^D(r) = [\omega^R(r)]^2 = \begin{cases} \left(1 - \frac{r}{R_c}\right)^2 & (r \leq R_c) \\ 0 & (r > R_c) \end{cases} \quad (61)$$

In the original DPD model, one critical advantage is the soft repulsive nature of conservative potential, which enables us to integrate the equations of motion with large time step. However, such advantage restricts the direct incorporation of electrostatic interactions in DPD model. The main problem is that dissipative particles carrying opposite point charges tend to collapse onto each other, forming artificial ionic clusters due to the stronger electrostatic interactions than soft repulsive conservative interactions.

In order to avoid such unphysical phenomena, point charges at the center of dissipative particles are usually replaced by charge density distributions meshed around particles to remove the divergency of electrostatic interactions at  $r=0$  [69,70]. In our implementation [71,72], a Slater-type charge density distribution is considered with the form of

$$\rho_e(r) = \frac{q}{\pi\lambda_e^3} e^{-\frac{2r}{\lambda_e}}, \quad (62)$$

in which  $\lambda_e$  is the decay length. The integration of Equation (62) over the whole space gives the total charge  $q$ .

The electrostatic potential  $\phi(r)$  generated by Slater-type charge distribution  $\rho_e(r)$  can be obtained by solving Poisson's equation

$$\nabla^2\phi(r) = -\frac{1}{\epsilon_0\epsilon_r}\rho_e(r), \quad (63)$$

in which the variables  $\epsilon_0$  and  $\epsilon_r$  are the permittivity of vacuum space and the dielectric constant of water at room temperature, respectively. In spherical coordinates, the Poisson's equation becomes

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial}{\partial r} \phi(r) \right) = -\frac{1}{\epsilon_0\epsilon_r} \rho_e(r). \quad (64)$$

By solving Poisson's equation, the electrostatic potential field  $\phi(r)$  can be analytically expressed by

$$\phi(r) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q}{r} \left( 1 - \left( 1 + \frac{r}{\lambda_e} \right) e^{-\frac{2r}{\lambda_e}} \right). \quad (65)$$

The electrostatic energy between two interacting charge density distributions  $i$  and  $j$  is the product of the charge density distribution  $i$  and the electrostatic potential generated by charge distribution  $j$  at position  $r_i$

$$\mathbf{U}_{qq}^{E,DPD}(r_{ij}) = q_i\phi_j(r_i) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{r_{ij}} \left( 1 - \left( 1 + \frac{r_{ij}}{\lambda_e} \right) e^{-\frac{2r_{ij}}{\lambda_e}} \right). \quad (66)$$

The electrostatic force on charge distribution  $i$  is the negative of the derivative of the potential energy  $\mathbf{U}_{qq}^{E,DPD}$  respect to its position  $r_i$

$$\begin{aligned} \mathbf{F}_i^{E,DPD}(r_{ij}) &= -\nabla_i \mathbf{U}_{qq}^{E,DPD}(r_{ij}) \\ &= \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{(r_{ij})^2} \left\{ 1 - e^{-\frac{2r_{ij}}{\lambda_e}} - \frac{2r_{ij}}{\lambda_e} e^{-\frac{2r_{ij}}{\lambda_e}} - \frac{2r_{ij}^2}{\lambda_e^2} e^{-\frac{2r_{ij}}{\lambda_e}} \right\}. \end{aligned} \quad (67)$$

By defining parameter  $r^* = r/R_c$  as the reduced center-to-center distance between two charge distributions and dimensionless parameter  $\beta = R_c/\lambda_e$ , respectively, the reduced electrostatic energy and force between two Slater-type charge distributions are given by



$$U_{qq}^{E,DPD}(r_{ij}^*) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{(R_c r_{ij}^*)^1} \left\{ 1 - (1 + \beta r_{ij}^*) e^{-2\beta r_{ij}^*} \right\}, \quad (68)$$

$$\mathbf{F}_i^{E,DPD}(r_{ij}^*) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{(R_c r_{ij}^*)^2} \left\{ 1 - (1 + 2\beta r_{ij}^* (1 + \beta r_{ij}^*)) e^{-2\beta r_{ij}^*} \right\}. \quad (69)$$

Comparing to the electrostatic energy and force between point charges in atomistic simulations in previous section, we can find that the electrostatic energy and force between two Slater-type charge distributions in DPD simulations are scaled with corresponding correction factors as

$$B_U = 1 - (1 + \beta r^*) e^{-2\beta r^*}, \quad (70)$$

$$B_F = 1 - (1 + 2\beta r^* (1 + \beta r^*)) e^{-2\beta r^*}. \quad (71)$$

The similarities between electrostatic energy and force between point charges and counterparts between charge density distributions imply that once we get electrostatic energy and force between point charges, from which the electrostatic energy and force between Slater-type charge density distributions in DPD simulations can be directly rescaled with corresponding correction factors.

For the electrostatic energy and force between point charges, we know that both of them do diverge when the relative distance between point charges is close to 0. While for the electrostatic energy and force between charge density distributions, in the limit of  $r_{ij}^* \rightarrow 0$ , the reduced electrostatic energy and force between charge density distributions are, respectively, described by

$$\lim_{r_{ij}^* \rightarrow 0} U_{qq}^{E,DPD}(r_{ij}^*) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{R_c} \beta, \quad (72)$$

$$\lim_{r_{ij}^* \rightarrow 0} \mathbf{F}_i^{E,DPD}(r_{ij}^*) = 0. \quad (73)$$

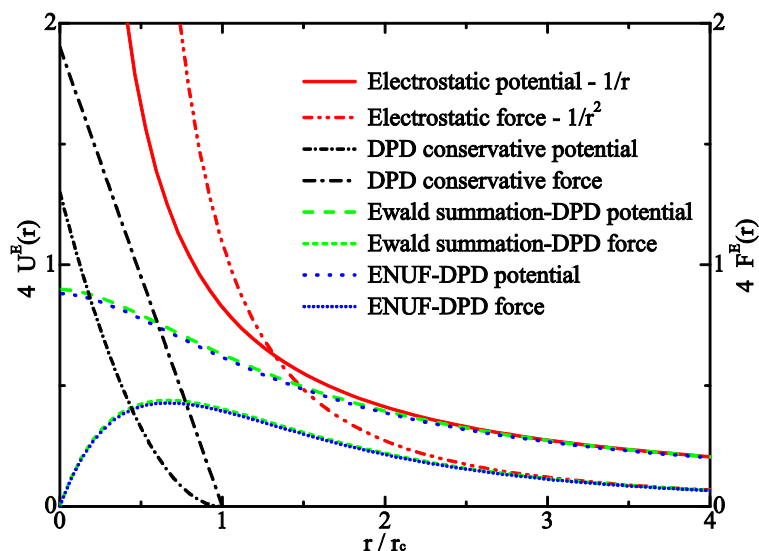
It is clear that the adoption of Slater-type charge distribution in DPD simulations removes the divergence of electrostatic interactions at  $r_{ij}^* = 0$ , which means that both electrostatic energy and force between charge distributions are characterized with finite quantities.

By matching the maximum electrostatic energy between charge distributions at  $r_{ij}^* = 0$  with Groot's previous work [69] gives  $\beta = 1.125$ . In our detailed implementations, we adopted a particular coarse-graining scheme [73] with  $N_m = 4$  and  $\rho = 4$ , in which the former parameter means 4 water molecules being coarse-grained into one DPD particle and the latter means there are 4 DPD particles in the volume of  $R_c^3$ . With this particular scheme, the length unit  $R_c$  is given as  $R_c = 3.107\sqrt[3]{\rho N_m} = 7.829\text{\AA}$ . From the relation of  $\beta = R_c/\lambda_e$ , we can get  $\lambda_e = 6.954\text{\AA}$ , which is consistent with the electrostatic smearing radii used in González-Melchor's work [70].

**Figure 2** shows the representation of reduced electrostatic energy and force with respect to the relative distance between Slater-type charge distributions. For a better comparison, we also include the typical soft conservative potential and force between dissipative particles, as well as the standard Coulombic potential and force between point charges, both of which do diverge at  $r = 0$ . In short range length scale, both the electrostatic energy and force are characterized with finite quantities, which attribute to the adoption of Slater-type charge distributions instead of point charges. In long range length scale, both the electrostatic energy and force are consistent with counterparts between point charges, which implies that the ENUF-DPD method can capture essential characteristics of electrostatic interactions at mesoscopic level.

Combining electrostatic force  $\mathbf{F}_i^{E,DPD}$  and soft repulsive conservative force  $\mathbf{F}_i^{C,DPD}$  gives the total conservative force  $\mathbf{F}_i^{C^*}$  on particles  $i$  in DPD simulations. The total conservative force  $\mathbf{F}_i^{C^*}$ , together with dissipative force  $\mathbf{F}_i^{D,DPD}$  and random force  $\mathbf{F}_i^{R,DPD}$ , as well as the intramolecular bonding force  $\mathbf{F}_i^{S,DPD}$  for polymers and surfactants, act on dissipative particles and evolve the whole simulated system toward equilibrium conditions before taking statistical analysis.

As the number of charged DPD particles in the simulated system grows, the calculation of the reciprocal space electrostatic interactions will become the most time-consuming part. Using suitable parameters in



**Figure 2.** Electrostatic potential and force between two charge density distributions in DPD scheme calculated from the ENUF and Ewald summation methods with reference parameters. For a better comparison, the standard Coulombic potential and force, both of which diverge at  $r = 0$ , and the typical conservative potential and force in standard DPD method are also included. Both the electrostatic potential and force expressions are plotted for two equal sign charge distributions.

ENUF-DPD method assures that the time to calculate the real space summations is approximately the same as the time to calculate the reciprocal space summations, thereby reducing the total computational time. Herein we try to explore the ENUF-DPD related parameters and get a set of suitable parameters for further applications.

As addressed in Section 3.2, the implementation of ENUF-DPD method uses the Ewald convergence parameter  $\alpha$ , required accuracy  $\delta (\ll 1)$ , and two cut-offs ( $r_{cut}$  for real space and  $n_{cut}$  for reciprocal space summations). These parameters are correlated with each other through two conditions shown in Equations (54) and (55). With required accuracy parameter  $\delta = 1.0 \times 10^{-4}$ , it is more convenient to pick a suitable value for  $n_{cut}$ . Then one can determine  $\alpha$  and  $r_{cut}$  directly from Equations (54). However, due to the fact that  $n_{cut}$  should be integer and  $r_{cut}$  should be a suitable value for the cell-link list update scheme in DPD simulations, we adopt another procedure to get suitable parameters.

During the calculation of real space summation and self-interaction parts of electrostatic energy and force between point charges, we can directly multiply corresponding correction factors to get the electrostatic energy and force between Slater-type charge density distributions. However, it is not accessible for the calculation of reciprocal space summation since we cannot rescale the NFFT transformation results. But if we choose suitable real space cutoff  $r_{cut}$ , beyond which two correction factors  $B_U$  and  $B_F$  converge to unit, one can directly adopt NFFT transformation result without any corrections. We find that electrostatic energy and force between charge density distributions are consistent with counterparts between point charges when the relative distance between two distributions is larger than  $3.0R_c$ . Hence in our simulations,  $r_{cut} = 3.0R_c$  is taken as the cutoff for real space summations of electrostatic interactions.

By evaluating the Madelung constant of a face-centered cubic (FCC) lattice, we adopt the Ewald convergence parameter with the value of  $\alpha = 0.20 \text{ \AA}^{-1}$ , which can generate accurate Madelung constant for FCC lattice structure and keep considerable accuracy. Then we perform coarse-grained simulations on bulk electrolyte system to explore suitable values for  $n_{cut}$ , and approximation parameter  $p$  in NFFT. It is specified that approximation parameter  $p = 2$  and cutoff  $n_{cut} = 7$  for reciprocal space summations can generate consistent electrostatic energy and force in comparison with those obtained from traditional Ewald summation method with reference parameters. Larger  $p$  values can further increase the accuracy of electrostatic interactions in ENUF-DPD method, but also the total computational time in treating electrostatic interactions increases. By compromising the accuracy and computational speed in the ENUF-DPD method, we adopt  $p = 2$  and  $n_{cut} = 7$  in following simulations.

With the set of explored optimized parameters, we address the computational complexity of ENUF-DPD method in treating electrostatic interactions. The computational complexity of ENUF-DPD method is appro-

ximately described as  $\mathcal{O}(N \log N)$ , which shows remarkably better computational efficiency than the traditional Ewald summation method with acceptable accuracy in treating long-range electrostatic interactions between charged particles at mesoscopic level.

The ENUF-DPD method is then validated by investigating the influence of charge fraction of polyelectrolyte on corresponding conformational properties [71]. With the increase of charge fraction on polyelectrolyte, both the intramolecular correlations between charged beads on polyelectrolyte chain and the intermolecular correlations between charged beads on polyelectrolyte and counterions are enhanced. The conformation transition of polyelectrolyte chain from collapsed state to fully extended conformation can be visualized from simulations. Meanwhile, the dependence of the conformations of fully ionized polyelectrolyte on charge valency and concentration of added salts are also studied in details. Counterions with larger valency show stronger condensations on polyelectrolyte chains. Such counterions can induce polyelectrolyte chains from extended conformation to compact state, and then to swollen conformation with the increase of counterion concentrations.

With the ENUF-DPD method, we further investigate the specific binding structures of dendrimers on amphiphilic bilayer membranes [74]. We construct mutually consistent coarse-grained models for dendrimers and lipid molecules, which can properly describe the conformation of charged dendrimers and the surface tension of amphiphilic membranes, respectively. Systematic simulations are performed and simulation results reveal that the permeability of dendrimers across membranes is enhanced upon increasing dendrimer sizes. The negative curvature of amphiphilic membrane formed in dendrimer-membrane complexes is related to dendrimer concentration. Higher dendrimer concentration together with the synergistic effect between charged dendrimers can also enhance the permeability of dendrimers across amphiphilic membranes.

With these two typical and representative applications, we can see that the newly implemented ENUF-DPD method can capture the essential characteristics of electrostatic interactions at mesoscopic level. This method has all capabilities of ordinary DPD method, but includes applications where electrostatic interactions are essential but previously inaccessible, hence can be used to study charged complex systems at mesoscopic level.

## 6. Summary and Conclusions

Treatment of electrostatic interactions based on Ewald summation techniques is reviewed. While Ewald-summation is still considered as the most accurate scheme to compute the long-ranged interactions, it is also the part slowing down simulations. The scaling for large systems makes the computations very time-consuming. In particular, the reciprocal part of Ewald becomes a bottle-neck. As an attractive alternative approach to mesh-based schemes which show a linear scaling, we introduce an Ewald method based on non-uniform fast Fourier transforms (ENUF) giving examples of two implementations in already existing software packages, for atomistic Molecular Dynamics and Dissipative Particle Dynamics. We demonstrate that the implementation becomes straight-forward as we rely on NFFT library. We discuss the optimization of convergence parameters and window functions.

The ENUF method scales linearly as  $\mathcal{O}(N \log N)$  and conserves both the energy and momentum to float point accuracy making it a very robust and accurate method.

## Acknowledgements

Y.-L. W. and A. L. acknowledge the KA Wallenberg Foundation for financial support. F. M. acknowledges the Wenner-Gren and the Carl Tryggers Foundations for funding for Visiting Professorship. Y.-L. W. and A. L. thank the Kavli Institute for Theoretical Physics China (KITPC) and Chinese Academy of Sciences (CAS) for support and scientific environment during their stay. This work is supported by SERC the Swedish e-Science Center and Swedish Science Council. Parts of the computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at PDC, HPC2N, and NSC.

## REFERENCES

- [1] M. P. Allen and D. J. Tildesley, "Computer Simulation of Liquids," Oxford University Press, New York, 1989.
- [2] J. M. Haile, "Molecular Dynamics Simulation: Elementary Methods," John Wiley & Sons, New York, 1992.
- [3] D. Frenkel and B. Smit, "Understanding Molecular Simulation: From Algorithms to Applications," Academic Press, Inc., Orlando, 1996.
- [4] D. M. York, T. A. Darden and L. G. Pedersen, "The Effect of Long-Range Electrostatic Interactions in Simulations of Macro-

- molecular Crystals: A Comparison of the Ewald and Truncated List Methods,” *Journal of Chemical Physics*, Vol. 99, No. 10, 1993, p. 8345. <http://dx.doi.org/10.1063/1.465608>
- [5] M. Bergdorf, C. Peter and P. H. Hünenberger, “Influence of Cut-off Truncation and Artificial Periodicity of Electrostatic Interactions in Molecular Simulations of Solvated Ions: A Continuum Electrostatics Study,” *Journal of Chemical Physics*, Vol. 119, No. 17, 2003, p. 9129. <http://dx.doi.org/10.1063/1.1614202>
- [6] C. Anézo, A. H. de Vries, H. D. Höltje, D. P. Tieleman and S. J. Marrink, “Methodological Issues in Lipid Bilayer Simulations,” *Journal of Physical Chemistry B*, Vol. 107, No. 35, 2003, pp. 9424-9433. <http://dx.doi.org/10.1021/jp0348981>
- [7] M. Patra, M. Karttunen, M. T. Hyvönen, E. Falck, P. Lindqvist and I. Vattulainen, “Molecular Dynamics Simulations of Lipid Bilayers: Major Artifacts Due to Truncating Electrostatic Interactions,” *Biophysical Journal*, Vol. 84, No. 6, 2003, pp. 3636-3645. [http://dx.doi.org/10.1016/S0006-3495\(03\)75094-2](http://dx.doi.org/10.1016/S0006-3495(03)75094-2)
- [8] P. P. Ewald, “Die Berechnung Optischer und Elektrostatischer Gitterpotentiale,” *Annalen der Physik*, Vol. 369, No. 3, 1921, pp. 253-287. <http://dx.doi.org/10.1002/andp.19213690304>
- [9] S. W. de Leeuw, J. W. Perram and E. R. Smith, “Simulation of Electrostatic Systems in Periodic Boundary Conditions, I. Lattice Sums and Dielectric Constants,” *Proceedings of the Royal Society London, Series A*, Vol. 373, No. 1752, 1980, pp. 27-56. <http://dx.doi.org/10.1098/rspa.1980.0135>
- [10] D. M. Heyes, “Electrostatic Potentials and Fields in Infinite Point Charge Lattices,” *Journal of Chemical Physics*, Vol. 74, No. 3, 1981, pp. 1924-1929.
- [11] T. Darden, D. York and L. Pedersen, “Particle Mesh Ewald: An  $M\log N$  Method for Ewald Sums in Large Systems,” *Journal of Chemical Physics*, Vol. 98, No. 12, 1993, pp. 10089-10092. <http://dx.doi.org/10.1063/1.464397>
- [12] D. York and W. Yang, “The Fast Fourier Poisson Method for Calculating Ewald Sums,” *Journal of Chemical Physics*, Vol. 101, No. 4, 1994, pp. 3298-3300. <http://dx.doi.org/10.1063/1.467576>
- [13] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee and L. G. Pedersen, “A Smooth Particle Mesh Ewald Method,” *Journal of Chemical Physics*, Vol. 103, No. 19, 1995, pp. 8577-8593. <http://dx.doi.org/10.1063/1.470117>
- [14] P. F. Batcho and T. Schlick, “New Splitting Formulations for Lattice Summations,” *Journal of Chemical Physics*, Vol. 115, No. 18, 2001, p. 8312. <http://dx.doi.org/10.1063/1.1412247>
- [15] D. R. Wheeler and J. Newman, “A Less Expensive Ewald Lattice Sum,” *Chemical Physics Letters*, Vol. 366, No. 5-6, 2002, pp. 537-543. [http://dx.doi.org/10.1016/S0009-2614\(02\)01644-5](http://dx.doi.org/10.1016/S0009-2614(02)01644-5)
- [16] Y. Shan, J. L. Klepeis, M. P. Eastwood, R. O. Dror and D. E. Shaw, “Gaussian Split Ewald: A Fast Ewald Mesh Method for Molecular Simulation,” *Journal of Chemical Physics*, Vol. 122, No. 5, 2005, Article ID: 054101. <http://dx.doi.org/10.1063/1.1839571>
- [17] P. J. Steinbach and B. R. Brooks, “New Spherical-Cutoff Methods for Long-Range Forces in Macromolecular Simulation,” *Journal of Computational Chemistry*, Vol. 15, No. 7, 1994, pp. 667-683. <http://dx.doi.org/10.1002/jcc.540150702>
- [18] C. L. Brooks III, B. M. Pettitt and M. Karplus, “Structural and Energetic Effects of Truncating Long Ranged Interactions in Ionic and Polar Fluids,” *Journal of Chemical Physics*, Vol. 83, No. 11, 1985, p. 5897. <http://dx.doi.org/10.1063/1.449621>
- [19] K. S. Kim, “On Effective Methods to Treat Solvent Effects in Macromolecular Mechanics and Simulations,” *Chemical Physics Letters*, Vol. 156, No. 2-3, 1989, pp. 261-268. [http://dx.doi.org/10.1016/S0009-2614\(89\)87131-3](http://dx.doi.org/10.1016/S0009-2614(89)87131-3)
- [20] L. Greengard and V. Rokhlin, “A Fast Algorithm for Particle Simulations,” *Journal of Computational Physics*, Vol. 73, No. 2, 1987, pp. 325-348. [http://dx.doi.org/10.1016/0021-9991\(87\)90140-9](http://dx.doi.org/10.1016/0021-9991(87)90140-9)
- [21] J. E. Barnes and P. Hut, “A Hierarchical ( $M\log N$ ) Force-Calculation Algorithm,” *Nature*, Vol. 324, 1986, pp. 446-449. <http://dx.doi.org/10.1038/324446a0>
- [22] J. Pérez-Jordá and W. Yang, “A Simple  $O(M\log N)$  Algorithm for the Rapid Evaluation of Particle-Particle Interactions,” *Chemical Physics Letters*, Vol. 247, No. 4-6, 1995, pp. 484-490.
- [23] Z. H. Duan and R. Krasny, “An Adaptive Treecode for Computing Nonbonded Potential Energy in Classical Molecular Systems,” *Journal of Computational Chemistry*, Vol. 22, No. 2, 2001, pp. 184-195. [http://dx.doi.org/10.1002/1096-987X\(20010130\)22:2<184::AID-JCC6>3.0.CO;2-7](http://dx.doi.org/10.1002/1096-987X(20010130)22:2<184::AID-JCC6>3.0.CO;2-7)
- [24] I. Tsukerman, “Efficient Computation of Long-Range Electromagnetic Interactions without Fourier Transforms,” *IEEE Transactions on Magnetics*, Vol. 40, No. 4, 2004, pp. 2158-2160. <http://dx.doi.org/10.1109/TMAG.2004.829022>
- [25] E. T. Ong, K. M. Lim, K. H. Lee and H. P. Lee, “A Fast Algorithm for Three-Dimensional Potential Fields Calculation: Fast Fourier Transform on Multipoles,” *Journal of Computational Physics*, Vol. 192, No. 1, 2003, pp. 244-261. <http://dx.doi.org/10.1016/j.jcp.2003.07.004>
- [26] C. Sagui and T. Darden, “Multigrid Methods for Classical Molecular Dynamics Simulations of Biomolecules,” *Journal of Chemical Physics*, Vol. 114, No. 15, 2001, p. 6578. <http://dx.doi.org/10.1063/1.1352646>
- [27] R. D. Skeel, I. Tezcan and D. J. Hardy, “Multiple Grid Methods for Classical Molecular Dynamics,” *Journal of Computational Chemistry*, Vol. 23, No. 6, 2002, pp. 673-684. <http://dx.doi.org/10.1002/jcc.10072>
- [28] J. A. Izaguirre, S. S. Hampton and T. Matthey, “Parallel Multigrid Summation for the N-Body Problem,” *Journal of Parallel*

- and *Distributed Computing*, Vol. 65, No. 8, 2005, pp. 949-962. <http://dx.doi.org/10.1016/j.jpdc.2005.03.006>
- [29] J. A. Baker and R. O. Watts, "Monte Carlo Studies of the Dielectric Properties of Water-Like Models," *Molecular Physics*, Vol. 26, No. 3, 1973, pp. 789-792. <http://dx.doi.org/10.1080/00268977300102101>
- [30] I. G. Tironi, R. Sperb, P. E. Smith and W. F. van Gunsteren, "A Generalized Reaction Field Method for Molecular Dynamics Simulations," *Journal of Chemical Physics*, Vol. 102, No. 13, 1995, p. 5451. <http://dx.doi.org/10.1063/1.469273>
- [31] T. N. Heinz and P. H. Hünenberger, "Combining the Lattice-Sum and Reaction-Field Approaches for Evaluating Long-Range Electrostatic Interactions in Molecular Simulations," *Journal of Chemical Physics*, Vol. 123, No. 3, 2005, Article ID: 034107. <http://dx.doi.org/10.1063/1.1955525>
- [32] R. W. Hockney and J. W. Eastwood, "Computer Simulation Using Particles," McGraw-Hill, New York, 1981.
- [33] B. A. Luty, M. E. Davis, I. G. Tironi and W. F. van Gunsteren, "A Comparison of Particle-Particle, Particle-Mesh and Ewald Methods for Calculating Electrostatic Interactions in Periodic Molecular Systems," *Molecular Simulation*, Vol. 14, No. 1, 1994, pp. 11-20. <http://dx.doi.org/10.1080/08927029408022004>
- [34] X. Wu and B. R. Brooks, "Isotropic Periodic Sum: A Method for the Calculation of Long-Range Interactions," *Journal of Chemical Physics*, Vol. 122, No. 4, 2005, Article ID: 044107. <http://dx.doi.org/10.1063/1.1836733>
- [35] T. M. Apostol, "Mathematical Analysis," Addison-Wesley, Boston, 1979.
- [36] J. W. Perram, H. G. Petersen and S. W. de Leeuw, "An Algorithm for the Simulation of Condensed Matter Which Grows as the  $3/2$  Power of the Number of Particles," *Molecular Physics*, Vol. 65, No. 4, 1988, pp. 875-893. <http://dx.doi.org/10.1080/00268978800101471>
- [37] J. Kolafa and J. W. Perram, "Cutoff Errors in the Ewald Summation Formulae for Point Charge Systems," *Molecular Simulation*, Vol. 9, No. 5, 1992, pp. 351-368. <http://dx.doi.org/10.1080/08927029208049126>
- [38] S. W. de Leeuw, J. W. Perram and E. R. Smith, "Simulation of Electrostatic Systems in Periodic Boundary Conditions. II. Equivalence of Boundary Conditions," *Proceedings of the Royal Society London, Series A*, Vol. 373, No. 1752, 1980, pp. 57-66. <http://dx.doi.org/10.1098/rspa.1980.0136>
- [39] S. W. de Leeuw, J. W. Perram and E. R. Smith, "Simulation of Electrostatic Systems in Periodic Boundary Conditions. III. Further Theory and Applications," *Proceedings of the Royal Society London, Series A*, Vol. 388, No. 1794, 1983, pp. 177-193. <http://dx.doi.org/10.1098/rspa.1983.0077>
- [40] W. H. Press and G. B. Rybicki, "Fast Algorithm for Spectral Analysis of Unevenly Sampled Data," *Astrophysical Journal*, Vol. 338, 1989, pp. 277-280. <http://dx.doi.org/10.1086/167197>
- [41] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, Vol. 19, 1965, pp. 297-301. <http://dx.doi.org/10.1090/S0025-5718-1965-0178586-1>
- [42] A. Brandt, "Multilevel Computations of Integral Transforms and Particle Interactions with Oscillatory Kernels," *Computer Physics Communications*, Vol. 65, No. 1-3, 1991, pp. 24-38. [http://dx.doi.org/10.1016/0010-4655\(91\)90151-A](http://dx.doi.org/10.1016/0010-4655(91)90151-A)
- [43] M. Pippig and D. Potts, "Parallel Three-Dimensional Nonequispaced Fast Fourier Transforms and Their Application to Particle Simulation," *SIAM Journal on Scientific Computing*, Vol. 35, No. 4, 2013, pp. 411-437. <http://dx.doi.org/10.1137/120888478>
- [44] O. Ayala and L. P. Wang, "Parallel Implementation and Scalability Analysis of 3D Fast Fourier Transform Using 2D Domain Decomposition," *Parallel Computing*, Vol. 39, No. 1, 2013, pp. 58-77. <http://dx.doi.org/10.1016/j.parco.2012.12.002>
- [45] Q. Liu and N. Nguyen, "An Accurate Algorithm for Nonuniform Fast Fourier Transforms (NUFFT's)," *IEEE Microwave and Guided Wave Letters*, Vol. 8, No. 1, 1998, pp. 18-20. <http://dx.doi.org/10.1109/75.650975>
- [46] G. Steidl, "A Note on Fast Fourier Transforms for Nonequispaced Grids," *Advances in Computational Mathematics*, Vol. 9, No. 3-4, 1998, pp. 337-352. <http://dx.doi.org/10.1023/A:1018901926283>
- [47] J. A. Fessler and B. P. Sutton, "Nonuniform Fast Fourier Transforms Using Min-Max Interpolation," *IEEE Transactions on Signal Processing*, Vol. 51, No. 2, 2003, pp. 560-574. <http://dx.doi.org/10.1109/TSP.2002.807005>
- [48] C. Anderson and M. D. Dahleh, "Rapid Computation of the Discrete Fourier Transform," *SIAM Journal on Scientific Computing*, Vol. 17, No. 4, 1996, pp. 913-919. <http://dx.doi.org/10.1137/0917059>
- [49] A. Duijndam and M. Schonewille, "Nonuniform Fast Fourier Transform," *Geophysics*, Vol. 64, No. 2, 1999, p. 539. <http://dx.doi.org/10.1190/1.1444560>
- [50] J. P. Boyd, "A Fast Algorithm for Chebyshev, Fourier, and Sinc Interpolation onto an Irregular Grid," *Journal of Computational Physics*, Vol. 103, No. 2, 1992, pp. 243-257. [http://dx.doi.org/10.1016/0021-9991\(92\)90399-J](http://dx.doi.org/10.1016/0021-9991(92)90399-J)
- [51] A. F. Ware, "Fast Approximate Fourier Transforms for Irregularly Spaced Data," *SIAM Review*, Vol. 40, No. 4, 1998, pp. 838-856. <http://dx.doi.org/10.1137/S003614459731533X>
- [52] A. Nieslony and G. Steidl, "Approximate Factorizations of Fourier Matrices with Nonequispaced Knots," *Linear Algebra and its Applications*, Vol. 366, 2003, pp. 337-351. [http://dx.doi.org/10.1016/S0024-3795\(02\)00496-2](http://dx.doi.org/10.1016/S0024-3795(02)00496-2)
- [53] J. J. Benedetto and P. J. Ferreira, "Modern Sampling Theory: Mathematics and Applications," Springer, Berlin, 2001. <http://dx.doi.org/10.1007/978-1-4612-0143-4>

- [54] S. Kunis and D. Potts, “NFFT3.2.3,” Institute of Mathematics, University of Lübeck, D-23560 Lübeck, 2013. <http://www.math.uni-luebeck.de/potts/nfft/>
- [55] F. Hedman and A. Laaksonen, “A Data-Parallel Molecular Dynamics Method for Liquids with Coulombic Interactions,” *Molecular Simulation*, Vol. 14, No. 4-5, 1995, pp. 235-244. <http://dx.doi.org/10.1080/08927029508022020>
- [56] F. Hedman, “Algorithms for Molecular Dynamics Simulations,” Ph.D. Thesis, Stockholm University, Stockholm, 2006. <http://www.diva-portal.org/smash/record.jsf?searchId=1&pid=diva2:178477>
- [57] F. Hedman and A. Laaksonen, “Ewald Summation Based on Nonuniform Fast Fourier Transform,” *Chemical Physics Letters*, Vol. 425, No. 1-3, 2006, pp. 142-147. <http://dx.doi.org/10.1016/j.cplett.2006.04.106>
- [58] M. Frigo and S. G. Johnson, “FFTW-The Design and Implementation of FFTW3,” *Proceedings of the IEEE*, Vol. 93, No. 2, 2005, pp. 216-231. <http://dx.doi.org/10.1109/JPROC.2004.840301>
- [59] A. P. Lyubartsev and A. Laaksonen, “M.DynaMix—A Scalable Portable Parallel MD Simulation Package for Arbitrary Molecular Mixtures,” *Computer Physics Communications*, Vol. 128, No. 3, 2000, pp. 565-589. [http://dx.doi.org/10.1016/S0010-4655\(99\)00529-9](http://dx.doi.org/10.1016/S0010-4655(99)00529-9)
- [60] J. P. Ryckaert, G. Ciccotti and H. J. C. Berendsen, “Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of n-Alkanes,” *Journal of Computational Physics*, Vol. 23, No. 3, 1977, pp. 327-341. [http://dx.doi.org/10.1016/0021-9991\(77\)90098-5](http://dx.doi.org/10.1016/0021-9991(77)90098-5)
- [61] M. Tuckerman, B. J. Berne and G. J. Martyna, “Reversible Multiple Time Scale Molecular Dynamics,” *Journal of Chemical Physics*, Vol. 97, 1992, pp. 1990-2001.
- [62] <http://www.yolinux.com/TUTORIALS/LinuxTutorialMixingFortranAndC.html>
- [63] A. P. Lyubartsev, M. DynaMix User Manual. [http://www.fos.su.se/sasha/md\\_prog.html](http://www.fos.su.se/sasha/md_prog.html)
- [64] P. J. Hoogerbrugge and V. A. Koelman, “Simulating Microscopic Hydrodynamic Phenomena with Dissipative Particle Dynamics,” *Europhysics Letters*, Vol. 19, No. 3, 1992, p. 155. <http://dx.doi.org/10.1209/0295-5075/19/3/001>
- [65] V. A. Koelman and P. J. Hoogerbrugge, “Dynamic Simulations of Hard-Sphere Suspensions under Steady Shear,” *Europhysics Letters*, Vol. 21, No. 3, 1993, p. 363. <http://dx.doi.org/10.1209/0295-5075/21/3/018>
- [66] Z. Y. Lu and Y. L. Wang, “An Introduction to Dissipative Particle Dynamics,” In: L. Monticelli and E. Salonen, Eds., *Methods in Molecular Biology: Biomolecular Simulations*, Humana Press, New York, 2013, pp. 617-633. [http://link.springer.com/protocol/10.1007/978-1-62703-017-5\\_24](http://link.springer.com/protocol/10.1007/978-1-62703-017-5_24)
- [67] P. Español and P. Warren, “Statistical Mechanics of Dissipative Particle Dynamics,” *Europhysics Letters*, Vol. 30, No. 4, 1995, p. 191. <http://dx.doi.org/10.1209/0295-5075/30/4/001>
- [68] R. D. Groot and P. B. Warren, “Dissipative Particle Dynamics: Bridging the Gap between Atomistic and Mesoscopic Simulation,” *Journal of Chemical Physics*, Vol. 107, No. 11, 1997, p. 4423. <http://dx.doi.org/10.1063/1.474784>
- [69] R. D. Groot, “Electrostatic Interactions in Dissipative Particle Dynamics-Simulation of Polyelectrolytes and Anionic Surfactants,” *Journal of Chemical Physics*, Vol. 118, No. 24, 2003, p. 11265. <http://dx.doi.org/10.1063/1.1574800>
- [70] M. González-Melchor, E. Mayoral, M. E. Velázquez and J. Alejandre, “Electrostatic Interactions in Dissipative Particle Dynamics Using the Ewald Sums,” *Journal of Chemical Physics*, Vol. 125, No. 22, 2006, p. 224107. <http://dx.doi.org/10.1063/1.2400223>
- [71] Y. L. Wang, A. Laaksonen and Z. Y. Lu, “Implementation of Non-Uniform FFT Based Ewald Summation in Dissipative Particle Dynamics Method,” *Journal of Computational Physics*, Vol. 235, 2013, pp. 666-682. <http://dx.doi.org/10.1016/j.jcp.2012.09.023>
- [72] Y. L. Wang, “Electrostatic Interactions in Coarse-Grained Simulations: Implementations and Applications,” Ph.D. Thesis, Stockholm University, Stockholm, 2013. <http://www.diva-portal.org/smash/record.jsf?pid=diva2:641063>
- [73] A. AlSunaidi, W. K. den Otter and J. H. R. Clarke, “Liquid-Crystalline Ordering in Rod-Coil Diblock Copolymers Studied by Mesoscale Simulations,” *Philosophical Transactions of the Royal Society London A*, Vol. 362, No. 1821, 2004, pp. 1773-1781. <http://dx.doi.org/10.1098/rsta.2004.1414>
- [74] Y. L. Wang, Z. Y. Lu and A. Laaksonen, “Specific Binding Structures of Dendrimers on Lipid Bilayer Membranes,” *Physical Chemistry Chemical Physics*, Vol. 14, No. 23, 2012, pp. 8348-8359. <http://dx.doi.org/10.1039/c2cp40700k>