Incremental Algorithm for long range interactions

Semeho Edorh

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : Mathématiques et Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Semeho EDORH

Thèse dirigée par **Stephane REDON**, CR, INRIA

préparée au sein du **Laboratoire Jean Kuntzmann** dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Algorithmes incrémentaux pour le calcul des interactions longue portée

Incremental Algorithm for long range interactions

Thèse soutenue publiquement le **2 octobre 2018**,
devant le jury composé de :

Monsieur STEPHANE REDON
CHARGE DE RECHERCHE, INRIA CENTRE DE GRENOBLE RHÔNE-ALPES, Directeur de thèse
Monsieur OLIVIER COULAUD
DIRECTEUR DE RECHERCHE, INRIA CENTRE BORDEAUX - SUD-OUEST, Rapporteur
Monsieur MATTHIAS BOLTEN
PROFESSEUR, UNIVERSITE DE WUPPERTAL - ALLEMAGNE, Rapporteur
Monsieur JEAN-LOUIS BARRAT
PROFESSEUR, UNIVERSITE GRENOBLE ALPES, Président
Monsieur STEFANO MOSSA
INGENIEUR DE RECHERCHE, CEA GRENOBLE, Examinateur
Monsieur JERÔME MATHE
MAITRE DE CONFERENCES, UNIVERSITE D'EVRY-VAL D'ESSONNE, Examinateur

Remerciements

Il me sera très difficile de remercier tout le monde car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener cette thèse à son terme.

Je voudrais tout d'abord remercier grandement mon directeur de thèse, Stéphane Redon, pour tout son aide. Je suis ravi d'avoir travaillé en sa compagnie car outre son appui scientifique, il a toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse. Son enthousiasme, sa gentillesse, sa connaissance sur des sujets aussi divers que variés sont une infime partie de ce qui fait de lui le mentor idéal.

Olivier Coulaud et Matthias Bolten m'ont fait l'honneur d'être rapporteurs de ma thèse, ils ont pris le temps de m'écouter et de discuter avec moi. Leurs remarques m'ont permis d'envisager mon travail sous un autre angle. Pour tout cela je les remercie.

Je tiens à remercier Jérôme Mathé, Jean-Louis Barrat et Stephano Mossa pour avoir accepté de participer à mon jury de thèse.

Je tiens aussi à remercier Eric Soccorsi, Jean-Yves Richard, Jean-Michel Lemoine et Anicet Edarh de m'avoir encouragé dans mon cursus scientifique. Quatre rencontres inspirantes sur ces dix dernières années. Merci à eux d'avoir motivé mon goût pour la recherche.

Outre des <<*cakes*>> et quelques kilos en plus sur la balance, L'équipe NANO-D m'a apporté plus que des collègues. Tout d'abord, merci, Imma d'avoir été notre <<*maman*>>. Ton oreille attentive, tes conseils (et aussi nos prises de becs) ont été indispensables au bon déroulement de la thèse.

Je tiens aussi à remercier particulièrement Serguei Grudinin pour toutes nos discussions, ses conseils et son aide précieuse pour ma recherche bibliographique. Merci aussi de m'avoir fait souvent rire. J'attends toujours de gouter la <<*Russian Cuisine*>>.

Leonard Jaillet, merci de m'avoir accompagné tout au long de la thèse. Je t'en ai beaucoup demandé surtout en cette fin de thèse. Merci pour toutes tes remarques et corrections. Nous pouvons maintenant nous concentrer sur les futures parties endiablées de badminton.

Un grand merci à Guillaume, Dima, Yacine, et Amal pour votre soutien et aide dans l'amélioration de ma présentation. Désolé d'avoir squatté (un peu trop souvent) votre

bureau entre deux simulations. J'ai trouvé en Guillaume et Yacine deux passionnés, comme moi, de sport en général. Merci à Maria, Khoa et Krishna d'avoir été là pendant cette aventure. Merci à mes deux partenaires François et Alex avec qui j'ai partagé le bureau, les discussions philosophico-politiques, les soirées à la bobine, au Korner ou au PPM.

Et non! Je ne t'ai pas oublié Joss le boss. Merci pour tout Jocelyn.

Je remercie toutes les personnes que j'ai eu le bonheur de côtoyer chez NANO-D et que je n'ai pas eu la chance de remercier directement. Svetlana, la grande sœur et nos discussions philosophiques autour de <<Plus moche la life>> ; Zofia l'autre grande sœur ; Emilie ; Clément ; Marc et tous les autres.

D'un point de vue plus personnel, mes remerciements les plus profonds iront à ma youyou (Julie B.). Te rencontrer a été le <<*game-changer*>> de mon expérience grenobloise. Désolé pour t'avoir fait vivre un calvaire en cette fin de thèse. Mais bon c'est fini. J'aurai enfin un peu plus de temps pour nous les w/e (NDLR où il n'y a ni F1 ni Moto gp). Merci aussi au valet et à nos <<petits secrets en famille>>.

Merci à Catherine, à Marc et à toute la famille Bourget pour leur soutien indéfectible, leur accueil et leur réconfort.

Merci à mes amis et soutiens de longue date (Tati, Julie T. et D., Judith, Nadine, Lumen, Margaux, Matthieu, Colline, Raelle, Maud, Fab leflou...). Un grand merci à <<*mon gros*>> (Julie O.) et à nos dizaines de milliers d'heures sur Skype depuis 2012. Dans le même registre, merci à Serguei et à Yeena pour nos discussions interminables. L'océan atlantique n'a pas eu raison de notre amitié. Promis, on aura un peu plus de temps pour se battre dans TERA.

Cette thèse m'a permis de faire une multitude de rencontres amicales. Les lister toutes, prendrait sans doute, autant de pages que le manuscrit. Une pensée émue pour William, parti trop tôt. Merci à aleks pour notre année à jouer au tarot, à Melissa ma lascarde préférée, au panda Lisa, Eric, Flora et autres de Biomics/Genel. Merci à Julien R., Deborah (Loo), Romain, Pastu, Lucie, Clem, Nath, Maryam, Julien D. et tous les autres.

Enfin, merci pour votre soutien Daddy et maman malgré les 6500 kms qui nous séparent. J'espère vous avoir rendus fiers avec cette thèse. Merci à Kevin. Akpé na atavi Poupou, Lolo, Gustave, Yves, Navi Cé, Tassi akpé… Merci à vous de m'avoir accompagné et soutenu sur le chemin.

Kids, you tried your best and failed miserably. The lesson is: never try.

Homer Simpson

Abstract

Particle simulations have become an essential tool in various fields such as physics, astrophysics, biology, chemistry, climatology, and engineering, to name few. Usually, these computer simulations produce a temporal evolution of the system of interest by describing the motion of particles.

In order to perform reliable simulations, we must provide an accurate description of interaction forces undergone by each particle. In most cases, these forces mirror inter-particle interactions and depend on relative coordinates of the particles. Moreover, pairwise long-range interactions are generally the cornerstone of particle simulations, an example being gravitational forces that are so essential in astrophysics. In molecular simulations, electrostatic forces are the most common illustration of long-range interactions.

Furthermore, due to their computational cost, pairwise long-range interactions are the bottleneck of particle simulations. Therefore, sophisticated algorithms must be used for efficient evaluations of these interactions.

In this thesis, we thus propose algorithms which may reduce the cost of long-range interactions when the studied system is governed by a particular dynamics. Precisely, these so-called «*incremental*» algorithms are effective for simulations where a part of the system remains frozen awhile. In particular, our algorithms will be validated on systems whose particles are governed by the so-called Adaptively Restrained Molecular Dynamics (ARMD) which is a promising approach in molecular dynamics simulations.

Although several incremental algorithms introduced by this thesis will be devoted to molecular dynamics simulations, we believe that they can be generalized to all kinds of long-range interactions.

Trying is the first step toward failure.

Homer Simpson

Nunya adidoe, asi ɖeka metunɛ O.

eve Lododowo

Résumé

Avec l'essor des ordinateurs, les simulations numériques sont devenues un outil de choix pour l'étude des systèmes de particules que l'on rencontre régulièrement en physique, en astrophysique, en biologie, en chimie, en climatologie, en ingénierie... Ces simulations fournissent le plus souvent une évolution temporelle du système d'étude en décrivant les mouvements des particules à des intervalles de temps (régulier ou non).

Afin de produire des simulations fiables, il est indispensable de fournir une description fidèle des forces d'interaction subies par chaque particule. De manière générale, ces forces décrivent l'influence des particules entre elles. Aussi, les forces d'interaction présentes en mécanique classique, dépendent le plus souvent de la position relative des particules.

Nous nous intéresserons au cas particulier des interactions dites à longue portée qui constituent la pierre angulaire de la plupart des simulations numériques de systèmes de particules. A titre d'exemple, ce sont les forces gravitationnelles omniprésentes en astrophysique. En dynamique moléculaire, les forces coulombiennes (ou électrostatiques) entre complexes chargés électriquement sont les interactions à longue portée les plus courantes.

Le coût exorbitant de l'évaluation de ces interactions constitue un facteur limitant pour la plupart des simulations numériques. Des algorithmes adaptés sont alors nécessaires pour un traitement efficient des interactions à longue portée. Dans cette optique, cette thèse propose des algorithmes « *incrémentaux* » qui peuvent réduire significativement le coût des interactions à longue portée lorsque le système d'étude a une dynamique singulière. En effet, ces algorithmes se montreront particulièrement efficaces lorsqu' une partie du système reste figée pendant un certain temps. Nous validerons ces algorithmes sur des systèmes dont l'évolution sera gouvernée par *la dynamique moléculaire dite restreinte de manière adaptative* (ARMD). Bien que les algorithmes incrémentaux introduits par cette thèse seront proposés pour la dynamique moléculaire, nous estimons qu'ils peuvent être étendus à n'importe quel type d'interactions à longue portée.

La science est comme le baobab.
Une seule main ne peut l'entourer.

Proverbe togolais

Table of Contents

List of Tables

List of Figures

Figure **Page**

Part I

Introduction and Mathematical tools

CHAPTER 1

A PERSPECTIVE ON PARTICLE SIMULATIONS

Contents

Johanes Kepler [1] demonstrated how colossal the amount of calculations required for a complete understanding of motions in a simple two-body system can be. He was the «*computer*» which provided a connection between Tycho Brahe empirical observations and Newton's theoretical models [15]. Four centuries later, it is astounding that human beings are able to easily study and understand systems that contain billion of particles. Particle simulations have experienced a resounding boom with the development of increasingly powerful computers. Since the 1950s, computer simulations have enabled, in many research fields, the study of large particle complexes more and more quickly. For instance, with the help of large supercomputers, the study of the formation of structures in the Universe becomes a major research area in astrophysics. This study resorts on simulations of systems involving a vast number of particles (*e.g.* more than 2 million massive collisionless particles were used in figure 1.1; 10 billion were used in the Millennium simulations [197]). Similarly, molecular simulations may involve several particles in order to examine the dynamics of atomic-level phenomena that cannot be observed directly (*e.g.* 0.5 million gold particles are melted in figure 1.2; more than 300 billion atoms have been simulated in [105]).

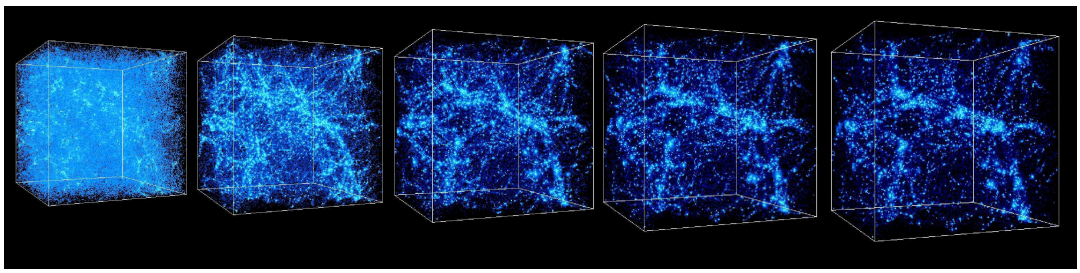


Figure 1.1: Computer simulations of formation of clusters and large-scale filaments in the Cold Dark Matter model with dark energy. Here, is shown the evolution of structures in a 43 million parsecs (or 140 million light years) box from redshift of 30 to the present epoch (left $z = 30$ to right $z = 0$). More details are available here [9] . Simulations were performed at the National Center for Supercomputer Applications by Andrey Kravtsov (The University of Chicago) and Anatoly Klypin (New Mexico State University). Visualizations by Andrey Kravtsov.

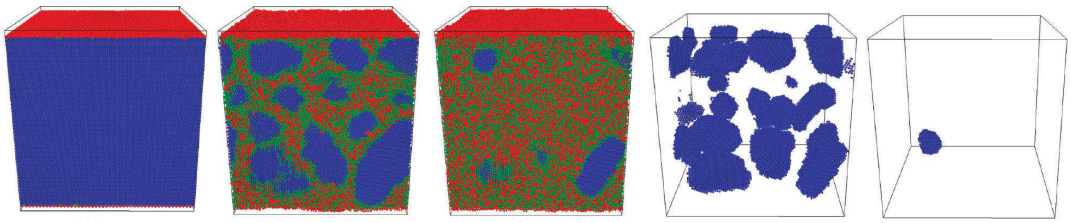


Figure 1.2: Molecular Dynamics Simulation of Laser Melting of Nanocrystalline Au. Snapshots of atomic configurations during the melting process in a $20nm$ nanocrystalline Au film irradiated with a 200 fs laser pulse at an absorbed fluence of $45\,\mathrm{J}^2\,\mathrm{m}^{-1}$ [122]

[1] Johannes Kepler (December 27, 1571 – November 15, 1630) was a German mathematician, astronomer, and astrologer, best known for his laws of planetary motion.

Although desktop computers can handle systems containing million of particles, billion of particles simulations are extremely demanding even for large clusters. For instance, in 2010, the «*Millennium XXL*» simulation (~ 303 billion particles) required more than 12000 cores for an equivalent of 300 years CPU time and 30 terabytes of RAM [116, 197].

In practice, particle simulations rely on a description of inter-particle interactions to derive the forces governing the motion for all particles. The determination of these interactions is usually the bottleneck of several particle simulations. Moreover, long-range pairwise interactions (especially gravitational and electrostatic interactions) are often the most expensive contributions. An efficient treatment of these interactions becomes a forefront research topic, especially in molecular simulations [10, 52]. The main focus of this thesis is in line with this thematic. For this reason, we first introduce several aspects of particle simulations.

1.1 Particle Simulation

1.1.1 The purpose

While using particle simulations, one generates several configurations (or states) of a given system in order to predict macroscopic properties (*e.g.* pressure, temperature, heat capacity). These simulations link microscopic description of the system to macroscopic informations. Statistical mechanics allow us to connect several macroscopic properties to long-time averages of microscopic states of the studied system [52]. For a practical point of view, average values obtained from particle simulations are often compared with empirical results in order to validate or interpret experiments.

In addition, averaged properties may provide a first glimpse of the studied problem and guide following empirical or theoretical investigations. In practice, particle simulations are constructed in order to match some prescribed experimental conditions. These conditions define some constraints based on conservation laws appropriate to the physical situation. They also characterize the set of all possible states of the system. In statistical mechanics, this set is called statistical ensemble. In practice, properties are retrieved by averaging over some states contained in the prescribed statistical ensemble. Below we list the commonly used ensembles [52].

- **Microcanonical ensemble** or (N, V, E) that describes the behavior of a thermally isolated system. In this ensemble the number of particles (N), the volume of the

system (V) and the total energy (E) are fixed. In addition, all accessible states have the same probability.

- **Canonical ensemble** or (N,V,T): Here, N,V and T are fixed. The temperature (T) is constrained with the help of a thermostat [92]. Precisely, the system is connected to a very large heat reservoir which dictates the temperature. The probability to reach a given state follows a Boltzmann distribution.

- **Isobaric-isothermal ensemble** or (N,P,T): Here, both the pressure of the system (P) and the temperature (T) are controlled. N,P and T are fixed.

- **Grand-Canonical ensemble** or (μ,V,T): Here, neither the total energy nor particle number are fixed. Instead, the chemical potential of the system (μ) , as well as the temperature (T) are constrained. μ, V and T are fixed and the number of particles can vary.

For this work, we will only consider microcanonical- and canonical- ensemble.

Given a statistical ensemble, one may generate the needed states by performing either Monte Carlo (MC) or Molecular Dynamics (MD) simulations [52, 158].

MC simulations generate a sequence of configurations as follows. From a given configuration (C_p), a trial configuration is created by randomly displacing some particles of the system. In reality, this new configuration is a candidate state that is accepted or rejected in accordance with an energy criterion. Precisely, the probability of acceptance is related to the difference in total energy between the new configuration and the old one. If the trial configuration is rejected, (C_p) is added (again) to the sequence. In case of an acceptance, the trial configuration is added to the sequence. The process is renewed starting from the retained configuration, here denoted by C_{p+1}.

On the other hand, Molecular Dynamics (MD) integrates the equations of motion of the system for some period of time. Starting from a set of initial conditions, we can predict the temporal evolution of a system of interacting particles [146]. By doing so, the temporal evolution of the system is generated [158]. This temporal evolution which is commonly called trajectory, represents a sequence of time-dependent configurations. We may thus average the desired property through this collection of configurations.

Under the «*ergodic hypothesis*», averaging over a sufficiently long period of time along MD trajectory is equivalent to the ensemble average generated with Monte Carlo simulations [52]. This claim is used intensively in particle simulations, even though it is neither proven nor always valid. For instance, when the generated trajectory is periodic,

there is no guarantee that the sampled configurations are sufficiently representative for a suitable computation of statistics. For this work, we will restrict ourselves to MD simulations. Nevertheless, several results are independent of the choice of the method employed to produce the system states.

At this point, we need to describe in more detail how the system's configurations are generated with the MD simulations. Therefore, we need to characterize more precisely the equations of motion.

1.1.2 Hamiltonian formalism

At a microscopic level, the description of constituent «particles» of matter (protons, neutrons, electrons...) requires the laws of quantum mechanics. Nonetheless, we restrict ourselves to its atomistic description. Thus, the matter consists of particles (*e.g.* atoms) which obey laws of classical mechanics. These particles interact with each other by forces which are usually described by potential energies originating from the quantum description or from empirical models. The broadened description of the behavior of such systems makes use of Hamiltonian formalism where each particle is characterized by its mass, position and momentum. Here, the momentum of a particle is equal to the product of its mass and its velocity.

Given a collection of N particles of masses $(m_1,\dots,m_N)$, let us denote by $\boldsymbol{q} = (\boldsymbol{q}_1,\dots,\boldsymbol{q}_N)$ the vector of all positions, by $\boldsymbol{p} = (\boldsymbol{p}_1,\dots,\boldsymbol{p}_N)$ the vector of all momenta. Usually, the Hamiltonian is understood as the total energy of the system, which is the sum of kinetic (K) and potential (V) energies. The time evolution[2] of the canonical coordinates $\boldsymbol{q}$ and $\boldsymbol{p}$ is governed by the following system of differential equations [11]:

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} &= \nabla_{\boldsymbol{p}} H \\ \dfrac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} &= -\nabla_{\boldsymbol{q}} H \end{cases} \tag{1.1}$$

where the initial conditions are $\boldsymbol{q}(0) = \boldsymbol{q}^0$ and $\boldsymbol{p}(0) = \boldsymbol{p}^0$.

We denote by H the Hamiltonian of the system under study. H often corresponds to the total energy of the system, which is the sum of kinetic (K) and potential (V) energies: $H = K + V$. The widely-used form of the Hamiltonian is separable, meaning that it can

[2] The reader should note that $\boldsymbol{q}$ and $\boldsymbol{p}$ are scleronomic (implicitly time-dependent).

be written in the form [75]

$$H(\boldsymbol{q},\boldsymbol{p}) = K(\boldsymbol{p})+V(\boldsymbol{q}) \tag{1.2}$$

$$= \frac{1}{2}\boldsymbol{p}^T M^{-1}\boldsymbol{p}+V(\boldsymbol{q}). \tag{1.3}$$

Here, we denote by M the $3N \times 3N$ diagonal matrix of masses

$$M = diag[m_1,\ldots,m_N]. \tag{1.4}$$

With respect to the above definition, the kinetic energy reads

$$K(\boldsymbol{p}) = \sum_{i=1}^{M} \frac{{p_i}^2}{2m_i}. \tag{1.5}$$

Although commonly used, this definition of kinetic energy is not unique. In particular, we will show in section 1.4 that a modified version of K can lead to some interesting dynamics. On the other hand, the expression of the potential energy V which describes the interactions between particles, is closely related to the physical model of the studied system. This leads to a variety of interaction potentials (*see* section 1.2). These interaction potentials are also called forcefields since they govern the forces acting on the particles. In fact, it comes with ease from (1.2) and (1.1) that

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla V \tag{1.6}$$

Then, the second Newton's law [153] of motions implies

$$\boldsymbol{f} = -\nabla V \tag{1.7}$$

where $\boldsymbol{f} = (\boldsymbol{f}_1,\ldots,\boldsymbol{f}_N)$ is a vector of all forces acting on particles under study.

1.1.3 Integration of equations of motion

Throughout MD simulations, the equations of motion (Equation (1.1)) are integrated. In most cases, these equations of motion cannot be solved analytically. Therefore, several numerical integration schemes have been developed [83]. From a practical point of view, an integration scheme generates a sequence of new configurations from a specified initial configuration. Each configuration represents the state (coordinates, velocities) of the studied system at a given step of integration.

Explicit integrators only use the current state of the system to determine the configuration of the next step. These integrators are simple to implement. However, some problems may be efficiently solved by implicit integrators which find a solution by involving both the current and the next states. Moreover, a special care is needed while choosing the integration timestep which separates two consecutive configurations. A too small timestep increases the cost of the integration, since several steps (and evaluations of the potential energy) are needed in order to access a configuration significantly distinct from the initial state. Conversely, a too large timestep may lead to an inaccurate and/or unstable integration. The accuracy can be improved by constructing the following configuration with the states computed at several previous steps (multi-step methods). In another way, one may improve the accuracy of the integration with the help of high-order integrators which are generally more expensive.

In order to preserve the geometric structure of the flow[3] of the Hamiltonian system (1.1), so-called symplectic integrators are available [65, 66, 93, 183]. When applied to Hamiltonian evolutions, symplectic schemes lead to a modified equation which remains Hamiltonian at all orders and allow long-term energy conservation. Symplectic schemes are the method of choice for integration of Hamilton's equations. However, time-reversible integrators can be employed since they share similar properties with symplectic schemes on some systems [83, 212].

1.1.4 Statistics

We emphasize that the aim of particle simulations is to provide a link between macroscopic properties and the microscopic description of a given system. This link is provided by averaging the desired property (also called observable) through an arbitrary statistical ensemble. As we only consider (N,V,E) and (N,V,T) ensembles, we can assume that the number of particles (N) and the volume of the system (V) are fixed. Moreover, a common way to sample this ensemble comes with MD simulations. Under the ergodic hypothesis, the time average of a given property A, obtained with a long enough MD simulation, is equal to its ensemble average, here denoted by $\langle A \rangle$.

The hamiltonian formulation defined with equations (1.1) and (1.2) can be used to describe microcanonical ensemble. First, it is important to notice that the employed Hamiltonian refers to the total energy of the system. Besides, we can show that this

[3] We recall that the flow of the Hamiltonian is an application ϕ_t which associates to some initial conditions the solution at t of the Hamiltonian system. $\phi_t : (\boldsymbol{q}^0, \boldsymbol{p}^0) \mapsto (\boldsymbol{q}(t), \boldsymbol{p}(t))$.

Hamiltonian is preserved by the dynamics [10]. Therefore, the total energy is constant along each MD trajectory and its value is fully determined by initial conditions.

Conversely, in (N,V,T) ensemble, the system we are interested in, passes through various energy levels. However, it has a well-defined temperature. The classical construction of a canonical ensemble embeds the studied system in an isolated reservoir. The system is maintained to a specified temperature by exchanging (solely) energy with the reservoir. The nature of these exchanges can be either deterministic or stochastic [164]. Often, deterministic approaches have natively some preserved quantities which can be monitored in order to validate the simulations. On the other hand, stochastic exchanges are easier to implement. Here, we will only consider Langevin dynamics [115] which is a stochastic dynamics that models a situation where a hamiltonian system is coupled to a thermostat[4]. Precisely, the hamiltonian system (1.1) is modified by introducing a stochastic perturbation. Thus, the dynamics is governed by the following stochastic differential equations [138]

$$\left\{\begin{array}{rcl} \dfrac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} &=& \nabla_{\boldsymbol{p}} H \\ \dfrac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} &=& -\nabla V - \gamma \nabla_{\boldsymbol{p}} H + \sigma \dfrac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} \end{array}\right. \tag{1.8}$$

where $\mathrm{d}\boldsymbol{W}$ is a standard $3N$-dimensional Wiener process which models a Brownian motion, γ and σ are $3N \times 3N$ matrices. $\gamma > 0$ is the friction which may depend on positions. For the sake of simplicity, the friction is considered as a positive constant. $\sigma \mathrm{d}\boldsymbol{W}$ is a fluctuation term which brings energy to the system. This energy is dissipated through a viscous friction $-\gamma \nabla_{\boldsymbol{p}} H = -\gamma M^{-1} p$. In order to ensure the preservation of the temperature T, the friction and the fluctuation terms are connected through the fluctuation-dissipation relation, namely,

$$\sigma \sigma^T = \frac{2\gamma}{\beta} \tag{1.9}$$

Moreover, the computations of average properties are often performed in (N,V,T) ensemble. These averages can be compared to empirical macroscopic properties that have been derived from experiments where the temperature can be easily monitored. Furthermore, the ensemble average of a given observable A in canonical ensemble reads

$$\langle A \rangle = \frac{\int A(\boldsymbol{q},\boldsymbol{p}) e^{-\beta H(\boldsymbol{q},\boldsymbol{p})} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}}{\int e^{-\beta H(\boldsymbol{q},\boldsymbol{p})} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}} \tag{1.10}$$

[4] Alternative approaches can be found in [52].

where $\beta = (k_B T)^{-1}$ denotes the thermodynamic beta and k_B is the Boltzmann constant. The above integrals run through all possible configurations [5] of $(\boldsymbol{q}, \boldsymbol{p})$. Each configuration is weighted by a so-called Boltzmann factor that depends on the related total energy – which is equal to $H(\boldsymbol{q}, \boldsymbol{p})$ with the definition (1.2).

1.2 Interaction potentials

In section 1.1.2, we showed that the motion of particles in the studied system can be linked to the interaction potential V (*see* equations (1.1) and (1.7)). Ideally, V should be derived from quantum mechanical computations based on an approximate numerical solution of the Schrödinger equation [37]. However, these quantum models are extremely demanding in terms of CPU resources. Therefore, popular molecular dynamics packages such as AMBER [219], CHARMM [35], CHARMM [19], LAMMPS [169] and NAMD [164] make use of computationally less expensive empirical potentials. In MD, a typical formulation of the potential energy splits the interactions into bonded and non-bonded contributions. Bonding terms are linked to interactions that hold some particles in a cohesive molecular structure. On the other hand, non-bonded contributions are mainly distance-dependent pairwise interactions. They are usually split into short-range (essentially induced by fast-decaying quantum interactions) and long-range (electrostatic and gravitational interactions) terms. Sometimes, external forces are applied to the system under study (*e.g* an external electric field). Given that, we can assume that the potential energy V can be expressed as follows

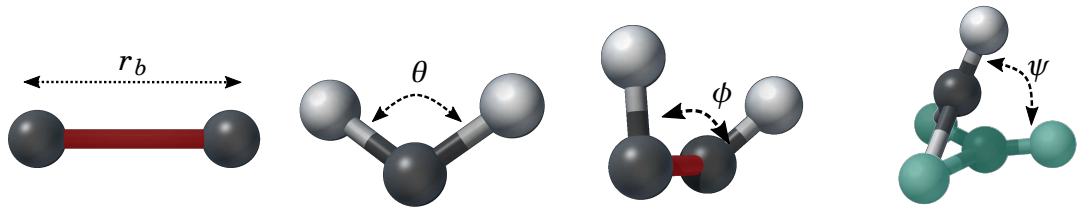
$$V = V_{bonded} + V_{short} + V_{long} + V_{ext}. \tag{1.11}$$

1.2.1 Bonding potentials

Bonding potentials are used in order to constrain the motion of covalently bonded particles. We recall that covalent bonds are lasting attractions between atoms that enable the formation of molecules through the sharing of electrons. Bonding potentials are usually split into bond-length, bond-angle, bond-torsion (or dihedral) and improper dihedral potentials:

$$V_{bonded} = \sum_{bonds} V_{bond}(r_b) + \sum_{angles} V_{angle}(\theta) + \sum_{dihedral} V_{dihedral}(\phi) + \sum_{improper} V_{improper}(\psi). \tag{1.12}$$

[5] The ensemble of all possible states is called phase space



(1) Bond length (2) Bond angle (3) Dihedral angle (4) Improper angle

Figure 1.3: Motions associated to bonded interactions

Bond-length potentials – These potentials control the length of covalent bonds. The **harmonic** potential is commonly used in order to describe vibrational motions between covalently bonded atoms. Based on the Hooke's law, this potential takes the form

$$V_{bond}(r_b) = k(r_b - r_0)^2 \tag{1.13}$$

where r_b denotes the length of a covalent bond, r_0 corresponds to the equilibrium separation, and k is the spring constant. While k is generally estimated with infrared adsorption and Raman spectroscopy, the equilibrium distance between atoms can be determined from X-ray diffraction experiments or using ab-initio quantum calculations [51]. The main drawback of the harmonic potential comes from its inability to capture the effects of bond breaking. The **Morse** potential is a convenient alternative when these effects are needed [215].

$$V_{bond}(r_b) = D_e \left[1 - e^{-S(r_b - r_0)}\right]^2 \tag{1.14}$$

Here, D_e is the dissociation energy and S controls the width of the potential well. Moreover, it interesting to remark that the Morse potential can be approximated by harmonic potential when r is close to r_0 (*see* figure 1.4).

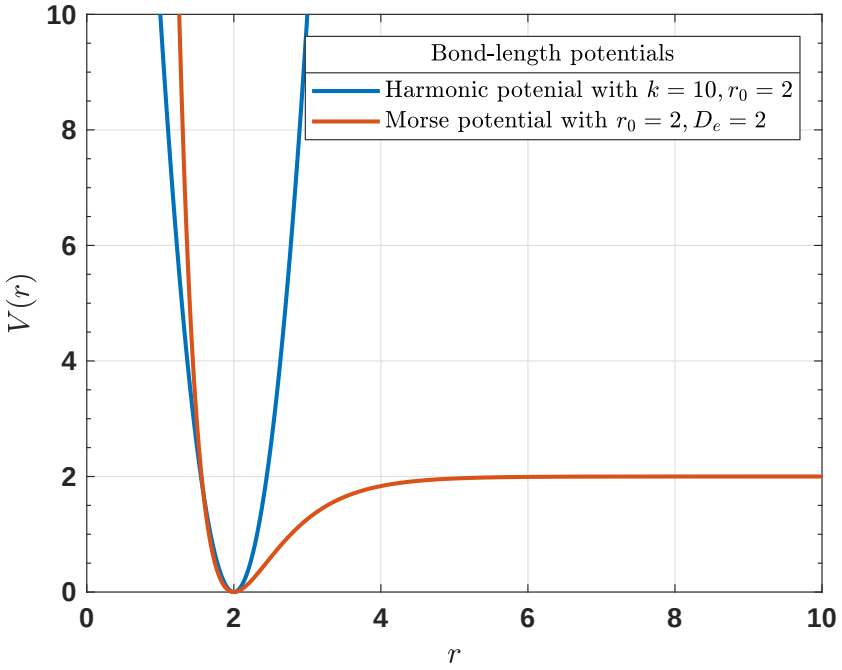


Figure 1.4: Comparison between Harmonic and Morse potentials for bond lengths.

Bond-angle potentials – These potentials govern the angle formed between three atoms across at least two bonds. The harmonic potential is a simple model for controlling the bond angles.

$$V_{angle}(\theta) = A(\theta - \theta_0)^2 \tag{1.15}$$

where θ_0 corresponds to the equilibrium angle, and A is the spring constant. Since it does not require the use of arccos function[6], the harmonic cosine (or trigonometric) potential is an efficient alternative to equation (1.15), namely,

$$V_{angle}(\theta) = A(\cos(\theta) - \cos(\theta_0))^2. \tag{1.16}$$

Nonetheless, several competitive variants of bond-angle potentials are still available in the literature [132, 139, 203].

Bond-torsion or dihedral angle potentials – These potentials are employed in order to impose the rotation around a given bond. The dihedral angle measures the angle between the planes formed by the first three and last three atoms of a consecutively

[6]Usually, simulations are performed with cartesian coordinates. Unlike the bond length which is directly available, one may need expensive arcus functions in order to retrieve angles. Conversely, one may use cosine laws in order to determine effortlessly the cosinus of angle.

bonded quadruple of atoms. A commonly used formulation of dihedral angle potential reads

$$V_{dihedral}(\phi) = \sum_m B_m \left[1 + \cos(m\phi - \phi_0)\right]. \tag{1.17}$$

Improper dihedral angle potentials – They enforce the planarity of certain parts (*e.g.* aromatic rings) or whole molecules (*e.g.* formaldehydes) and maintain chirality of some molecules [186]. Here again, the harmonic potential is a natural choice for improper angles:

$$V_{improper}(\psi) = C(\psi - \psi_0)^2. \tag{1.18}$$

1.2.2 Short-range potentials

Besides bonded potentials, atoms or molecules undergo several distance-dependent pairwise interactions. Some of those interactions quickly vanish at longer distances between interacting particles.

Given a collection of N particle, the short-range potential energy reads

$$V_{short} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j>i}}^{N} V_s(r_{ij}). \tag{1.19}$$

Here, V_s describes the pairwise interaction potential between two particles i and j. In particular, V_s depends on their separation r_{ij} and decay fast. The « fast (*resp.* slow) decay» property means that V_s goes down to zero faster (*resp.* slower) than r_{ij}^{-d}, where $d = 3$ the dimension of the problem.

Van der Waals forces[7] are the best known interactions that fall in this specific case. Although weaker than bonded potentials, van der Waals forces play a fundamental role in several fields such as surface science, structural biology, polymer science, nanotechnology, and condensed matter to name a few. A good understanding of the origin of these forces requires a quantum description of atomic interactions that unfortunately goes beyond the scope of this thesis. Nevertheless, it should be noted that van der Waals forces have four major contributions:

1. A repulsive component resulting from the Pauli exclusion principle that prevents the collapse of particles.

[7] named after Dutch scientist Johannes Diderik van der Waals

2. Keesom[8] forces which are attractive or repulsive electrostatic interactions between permanent charges or multipoles [109].

3. Debye[9] forces (also called polarization) which are attractive interactions between a permanent multipole on a given molecule with an induced multipole on another one.

4. London[10] dispersion interactions, which are attractive interactions induced by the non-zero instantaneous dipole moments of all atoms and molecules.

Furthermore, the LJ potential is an often employed approximate model for van der Waals interactions, namely,

$$V_{LJ}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{1.20}$$

where r the distance between two particles, σ represents the LJ-separation distance and ϵ is the depth of the LJ potential well.

Whilst the r^{-12} term describes the Pauli repulsion at small separations, the r^{-6} term describes weak attraction at larger distances. Moreover, this potential reaches its minimum for $r_{min} = 2^{1/6}\sigma \simeq 1.12\sigma$. As shown in figure 1.5, short potentials decay fast. Consequently, these potentials are generally truncated beyond a given cutoff distance r_c. For LJ potential, the cutoff distance is typically set at $r_c = 2.5\sigma$, since $V_{LJ}(r_c) \simeq \frac{1}{60} V_{LJ}(r_{min})$. The truncation is achieved by setting the potential to nil beyond r_c. Nevertheless, one must shift the truncated potential in order to cancel discontinuity induced by the truncation at r_c. By adding a so-called tail correction, we obtain the truncated and shifted LJ potential [21, 37], namely,

$$V^*_{LJ}(r_c) = \begin{cases} V_{LJ}(r) - V_{LJ}(r_c) & \text{when } r < r_c \\ 0 & \text{when } r \geq r_c \end{cases} \tag{1.21}$$

This model is broadly used in MD simulations as the definition equation (1.20) is computationally simple. In fact, interactions between particles further than the cutoff distance r_c, do not require any worthless computation.

Furthermore, the Lennard-Jones potential is extremely accurate for noble gas atoms (helium, neon, argon, krypton, xenon, radon, oganesson). More generally, it is a good

[8] named after the Dutch physicist Willem Hendrik Keesom.
[9] named after the Dutch-American physicist and physical chemist Peter J. W. Debye.
[10] named after the German-American physicist Fritz London.

approximation at long and short distances for neutral atoms and molecules. If necessary, more accurate (and thus more expensive) potentials are available in the literature [101]. For instance, the description of van der Walls interactions was significantly improved by Richard Buckingham who replaced the repulsive part of LJ potential by an exponential term [36, 101], namely,

$$V_B(r) = \gamma \left[\exp\left(\frac{r_0}{r}\right) - \left(\frac{r_0}{r}\right)^6 \right]. \tag{1.22}$$

Buckingham and LJ potentials are the common forms of the van der Walls interactions. In practice, parameters for van der Waals potentials can be estimated by fitting parameters to lattice energies and crystal structures, or from liquid simulations in order to reproduce empirical properties [48, 186].

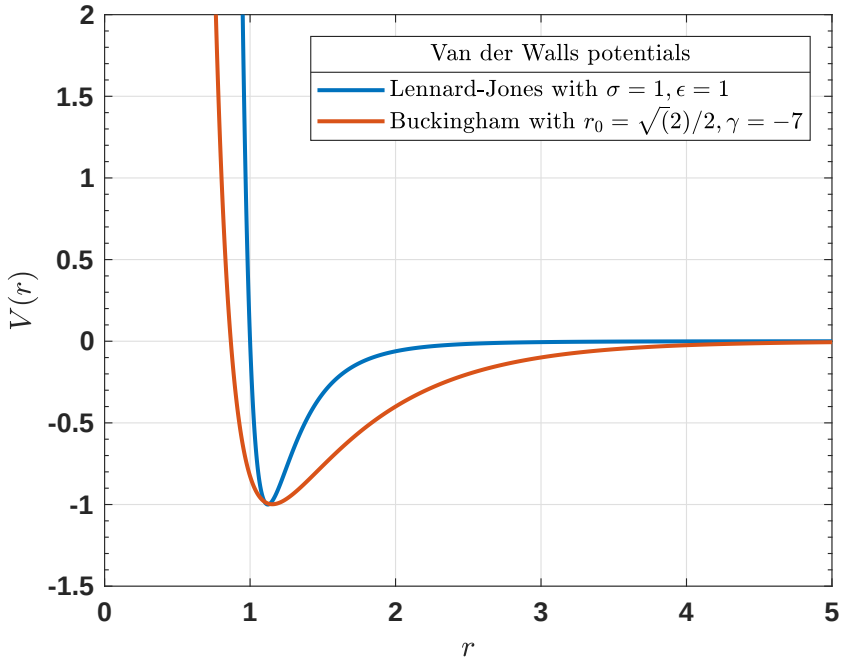


Figure 1.5: Comparison between Lennard-Jones and Buckingham potentials for van der Walls interactions.

1.2.3 Long-range potentials

Long-range interactions are extremely crucial for conducting meaningful particle simulations. Unfortunately, due to their slow decay, these potentials are computationally expensive. There are two main examples of long-range potentials:

Gravitational interactions are attractive forces induced by a distribution of masses, namely,

$$V_{grav} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j>i}}^{N} G \frac{m_i m_j}{r_{ij}} \tag{1.23}$$

where $G \simeq 6.674 \times 10^{-11} \mathrm{m}^3 \mathrm{kg}^{-1} \mathrm{s}^{-2}$ is the gravitational constant and r_{ij} denotes the inter-particle distance. Although gravitational forces are particularly omnipresent at the macroscopic level, astoundingly, they are negligible at the nanoscale.

Electrostatic interactions are induced by the distribution of charges and are result of attractive or repulsive interactions between fully or partially charged groups. Precisely, electronegative[11] atoms attract electrons more than less electronegative atoms. Consequently, the distribution of charge in a molecule is irregular. This charge distribution is often described with point charges localized throughout the molecule, mostly coinciding with the nuclei of the atoms. Given a collection of N point charges, the electrostatic or Coulomb energy reads

$$V_{elec} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j>i}}^{N} \frac{1}{4\pi\epsilon_r} \frac{q_i q_j}{r_{ij}}. \tag{1.24}$$

Beware, here q_i designates the charge of the i-th particle. ϵ_r denotes the permittivity of the medium.

In MD, electrostatics are needed in order to build more realistic models. For instance, they are important for stabilizing biomolecular conformations in solvent [186]. Unlike van der Walls interactions, long-range potentials cannot be neglected at far distance. Prior studies that have only counted theses interactions within a cutoff radius, have led to inaccuracies. On the other hand, evaluating all pairwise terms directly leads to an expensive $\mathcal{O}(N^2)$ computational complexity. Therefore, the efficient computation of long-range interactions, especially electrostatics, is the main focus of this thesis. Since the treatment of gravitational and electrostatic interactions are indeed similar, several results on the fast treatment of electrostatics (*see* chapters 3 and 4) can be extended with ease to both kinds of interactions.

[11] Electronegativity is a chemical property that describes the tendency of an atom to attract a shared pair of electrons (or electron density) towards itself.

1.2.4 The computational challenge

So far, we have described the building blocks of molecular dynamics simulations. From a practical point of view, a MD simulation can be broken down in a few steps. Starting from a given configuration, one evaluates the corresponding potential energy (V) which reflects the interactions arising in the system – V also is also related to forces acting on each particle in the system (equation (1.7)). Positions and momenta of particles are updated according to this potential energy (equation (1.1)). In this way, we can obtain a new configuration corresponding to the state of the simulated system a few moments later. In practice, only a few femtoseconds ($1\,\text{fs} = 10^{-15}\text{s}$) separate the novel configuration and its forerunner. In order to maintain the stability of the integrators, MD simulations are constrained to use short timestep size. From the newly generated configuration, one may calculate and/or output physical quantities of interest. Then, the process (also called step or timestep) is iterated as long as enough configurations are obtained.

In order to achieve sufficient precision in the computation of statistics, the simulation needs to capture relevant physical phenomena. Precisely, simulations must be conducted long enough to cover the characteristic time of fluctuations in the desired observable. In practice, MD simulations describe the evolution of a given system for a few period of real time (ranging from picosecond to microsecond regime [52]). Unfortunately, this corresponds to a large number of timesteps (usually $10^3 - 10^6$ MD steps), since the timestep size is extremely small. Depending on the size of the simulated system, each MD step can last several seconds (even minutes) in real (or wall-clock) time. Therefore, a successful simulation may require several days, week, or month of computations.

In practice, the bottleneck of molecular dynamics comes from the time spent evaluating the interaction potential. In particular, the computation of non-bonded contributions can be extremely time consuming since these pairwise interactions scale quadratically with the total number of particles. This scaling may be onerous for systems involving a large number of particles. On the other hand, we want to be able to perform long simulations leading to a large number of steps . Therefore, the reduction of the computational cost of the evaluation of the interaction potential is of paramount importance. Although we have reached the milestone of milliseconds simulations of biomolecular simulations [165, 166], the main challenge for MD simulations remains to improve their computational efficiency while retaining sufficient accuracy.

1.3 Efficient particle simulations

In order to overcome the limiting factor of MD, several strategies have been proposed since 1970s [52]. Some of those are presented in the following sections. They have proven their efficiency since the timescale of molecular simulations have been increased by a factor exceeding 10^8 [26].

The computational complexity of the interaction potential can be reduced by various means. First, we may decrease the dimensionality of the system by reducing the number of particles involved. The reduction can be achieved by altering the microscopic description of the system (*see* sections 1.3.1 and 1.3.2). On the other hand, one can optimize the way the interactions are computed by employing suitable data structures (*see* section 1.3.3). Furthermore, we can reduce the CPU time per evaluation of the interaction potential by performing approximate calculations (*see* sections 1.3.4 and 1.3.5).

Finally, several algorithms have been designed in order to take advantage of the parallel architecture of modern computers.

1.3.1 Particle representation

A molecular system can be represented in several ways. In quantum mechanics, one describes the system as a collection of atoms and subatomic particles characterized by wavefunctions. This description is extremely expensive, even for modern computers. Several simplifications have been proposed in order to able to simulate efficiently molecular systems (Figure 1.6(1)).

1.3.1.1 All-atom representation

The dimensionality of the studied system is reduced by using an atomistic representation. In many scientific fields, the molecular systems of interest can involve a large number of atoms. In practice, the size of these systems ranges from few thousands (*e.g* proteins) up to several millions of atoms (*e.g* viruses).

Moreover, a wide variety of empirical forcefields are employed alongside this model. Usually, these forcefields include several interaction potentials. In addition, they are parametrized for each atom type (*e.g* van der Waals cutoff, mass, charge...) and each bond type (equilibrium values for bond lengths, angles...) existing in the studied system. Rappé et al. [173] proposed a general force field that includes all types of atoms. Although this forcefield is useful when the studied system includes several atom types, most MD simulations limit themselves to more specific and less expensive forcefields [32, 131].

For instance, the brenner potential [32] is commonly used in the study of hydrocarbons. Alternatively, some approaches combine advantages of ab-initio and empirical potentials (*e.g* QM/MM [121]).

1.3.1.2 Atomic representation with reduced degrees of freedoms

Molecular simulations can be performed in the internal coordinates of a molecule (*i.e.* bond lengths, bond angles and torsion angles) [1, 81, 100, 175]. One may reduce the number of degrees of freedom in the system by fixing some coordinates (*e.g.* bond lengths or bond angles) to specific values. Although high- frequency fluctuations of the constrained coordinates are neglected, this approach has been successfully employed in several cases (*e.g.* NMR [81], molecular docking [1]). Nevertheless, it often happens that the derivation of the desired forcefield in terms of internal coordinates is neither trivial nor cheap.

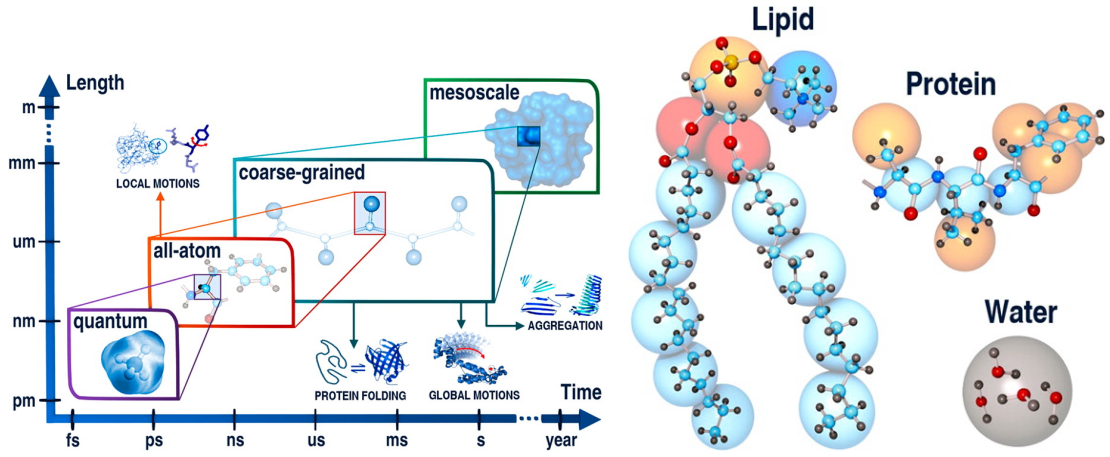
1.3.1.3 United-atom representation

The key idea of this model combines some hydrogen atoms with other atoms in order to create groups of atoms labeled as «pseudo-atoms». Thus, the number of particles to be considered decreases. Several forcefields have been dedicated to this model [103, 226].

1.3.1.4 Coarse-graining

Coarse-grained models extend the concept of pseudo-atoms of united-atom representation. Here, large groups of atoms such as water molecules are combined together. The coordinates of groups of atoms are mapped to new sites or pseudo-particles (Figure 1.6(2)). In addition, the interactions between pseudo-particles are determined by averaging over the fully detailed system. Coarse-grained models must redefine the number, types, and connectivity of the sites that are used to describe the system [155]. This approach which considerably reduces the number of particles, was first introduced by Levitt and Warshel [119]. Later, the method has been popularized by the MARTINI forcefield [136].

Coarse-graining enables simulations of much longer time-scales and/or larger systems. Thus, one may monitor global motions in the system. In particular, coarse-grained models are powerful and flexible tool for the prediction of protein interactions and for the study of protein folding[155]. A wide range of coarse-grained representations have been developed. The reader should refer to [111, 155] for an overview of coarse-grained protein models and their applications.



(1) Application ranges for molecular modeling at different resolutions

(2) All-atom versus coarse-grained representation in the MARTINI model

Figure 1.6: Application ranges for molecular modeling at different resolutions (Left). Sketch of the coarse-grained representation in the Martini model(right). Pictures taken from Kmiecik et al. [111]

Nowadays, several multiscale representations [106, 159, 171] have been proposed in order to combine all-atom and coarse-grained models. The studied system is described with different levels of granularity. This allows us to couple the efficiency of coarse-grained simulations and details of all-atom representations.

1.3.1.5 Water models

Water plays a major role in several MD simulations since it's often used as a solvent. Particles forming the solvent can represent more than 90% of atoms involved in the system (*e.g.* [113, 162]). A proper and efficient description of water is indeed essential for simulations in solutions.

Implicit solvent method [176] is often employed for describing water in MD simulations. This approach mimics the effect of the solvent by representing it as a continuous medium. Here, the solvent particles are not explicitly included in the system, leading to an extremely costless approach. The absence of explicit solvent particles induces a lack of viscosity since random collisions with solvent particles are not taken into account. Precisely, the motion of the solute particles is modified since the van der Walls repulsions induced by the solvent are ignored. Nonetheless, one can introduce the effects of the viscosity by modifying the dynamics with an frictional term. The so-called Langevin dynamics is often used to this end (*see* equation (1.8)). Implicit solvents have

been found useful in several studies [40, 94, 130, 229]. However, this approach is limited when more realistic effects is needed (*e.g* ionization, hydrophobic, hydrodynamic effects).

Alternatively, **explicit solvent method** explicitly treats the coordinates of particles that constitute the solvent. This approach provides a more accurate description of the water solvent while adding a significant computational cost. Furthermore, several explicit water models have been proposed. From a practical point of view, these models specify the geometry of a water molecule and constrain some molecular degrees of freedom.

- **Rigid models** are the simplest explicit water models as they treat the water molecule as rigid. Particles forming the water molecule are fixed in order to match a known geometry[12]. Thus, bonded interactions can be neglected and only non-bonded interactions (*e.g.* electrostatics and Lennard-Jones) have to be considered.
- **Flexible models** are used with the aim to reproduce molecular vibrations. Here, bond-length and bond-angle potentials are included.
- **Polarizable models** such as SPC/E [20], include explicit polarization term. Therefore, they can reproduce water in different phases and interaction between them.

In practice, most water models neglect the van der Walls repulsion of Hydrogen atoms. Moreover, the charged sites on the water molecule are placed on the atoms or on dummy sites (such as lone pairs). Therefore, water models are often classified in terms of p-sites, where p refers to the number of charged sites. In addition, they are parameterized in order to reproduce physical properties of water (*e.g.* density, heat of vaporization, diffusion constant, dielectric constant). Unfortunately, none of the available models is able to reproduce all properties of the water with good accuracy. As proof, the impressive amount of water models that have been proposed during five decades of computer simulations with water (46 models have been reviewed by Guillot [80]). Nonetheless, SPC [18], SPC/E [20], TIP3P, TIP4P [102] and TIP5P [133] are the commonly used water models.

In this manuscript, the employed water models are SPC/E in chapter 3, SPC/ϵ [69] in chapter 4 and TIP3P in chapter 5. Table 1.1 outputs parameters used in SPC/E model. Meanwhile, parameters of TIP3P and SPC/ϵ are provided in the appropriate chapters.

[12] which often corresponds to the average positions of particles in a water molecule at a specified temperaure.

Atom	charge (e)	mass (g/mol)	σ (Å)	ϵ (kcal/mol)
H	0.4238	1.0080	0.0000	0.0000
O	-0.8476	15.9994	3.1660	0.155354
Bond-length potential				
Harmonic	k (kcal/mol/Å^2)		r_0 (Å)	
O–H	1000		1.0	
Bond-angle potential				
Harmonic	A (kcal/mol/$^{\circ 2}$)		θ_0 ($^\circ$)	
H–O–H	100		109.47	

Table 1.1: Parameters of SPC/E water model. Lennard-Jones potential (σ, ϵ) is used for van der Walls interactions among oxygen atoms. Bonded interactions are modeled with harmonic bond-length and bond-angle potentials. Since SPC/E requires electrostatics, atomic partial charge are shown.

1.3.1.6 Polymer representation

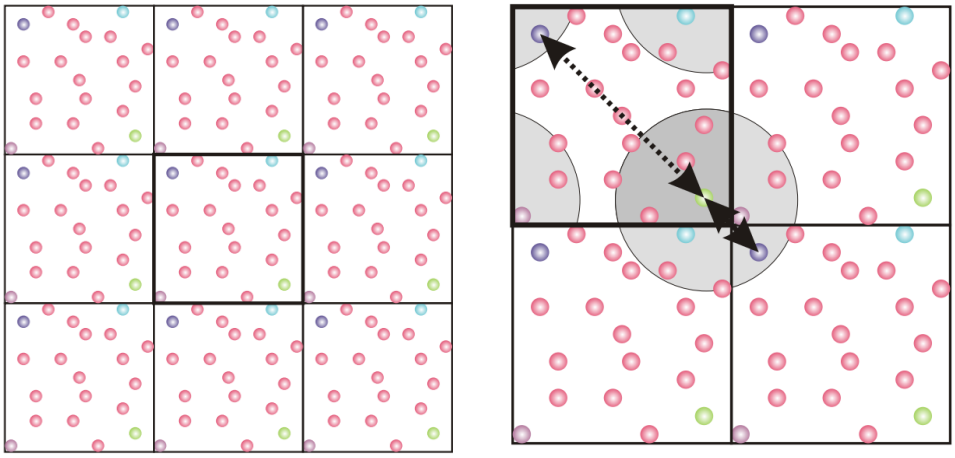
Due to their ubiquity in everyday life, polymers are often studied in molecular simulations. They are generally long chains of molecules made of repeated units of small molecules called monomers. In the so-called **polymerization** process, large numbers of monomers can be combined through covalent bonding. Often, the all-atom representation of polymer is too expensive for computer simulations. The bead-spring model is a cheaper but broadly adopted model for MD simulations of polymers. In this representation, monomers are represented as individual beads connected with harmonic springs. The interactions among these monomers are modeled by harmonic bond-length potential and non-bonded potentials. This bead-spring representation has been employed in chapter 5 where the translocation of a charged polymer (also called polyelectrolyte) is studied.

1.3.2 Periodic boundary conditions

Unsurprisingly, we cannot (yet) simulate large number of molecules which contains for instance $1\,\text{mole} \simeq 6 \times 10^{23}$ particles. However, the dimensionality of the problem can be reduced drastically with periodic boundary conditions (p.b.c) [52] which are commonly used in particle simulations. Under these conditions, large systems are represented by a small portion of particles forming a new sub-system which defines the main simulation

box. The simulation box is surrounded with replicas or periodic images. This way, the simulation box is artificially extended into infinite dimensional system. Astoundingly, this trick is extremely successful in computer simulations.

Under periodic boundary conditions, each particle interacts not only with its direct neighbors in the simulation box but also with all replicas. Traditionally, the minimum image convention is employed with p.b.c. With this convention, only interactions between the closest instance of a molecule are taken into consideration (*see* figure 1.7). This minimum image convention is extremely useful when accounting for short-ranged interactions. In fact, if the simulation box is chosen large enough, interactions with other replicas can be neglected.



(1) Periodic boundary conditions (2) Minimum image convention

Figure 1.7: Sketch of periodic boundary conditions. by considering its periodic images.

1.3.3 Neighbor lists

Thanks to p.b.c, the number of particles involved in molecular simulations can be reduced (to not more than few billions). So far, we reach a critical point where all bonded interactions that arise in the studied system, can be handled with ease. In fact, the number of bonded interactions required is the same order as the number of particles (bonds, angles...).

Conversely, the total number of non-bonded pairwise interactions (*e.g.* Lennard-Jones or electrostatics) is too large for computer simulations. In fact, the study of a system of N particles requires to account of at least $N(N-1)/2$ individual pairwise interactions. In molecular dynamics simulations, neighbor lists are widely used data structures that

efficiently maintain a list of all particles within a given cutoff distance of each other. With the help of neighbor lists, one can reduce the computational complexity of the computations of the evaluation of fast decaying pairwise interactions. We showed that in this specific case the interaction potential vanishes beyond a given cutoff r_c. In other words, two particles further than a certain cutoff radius, do not interact each other. While accounting of short-ranged potentials, each particle only interacts with its close neighbors. For each particle, one may hold the list of its neighbors (particles within this cutoff distance). Therefore, the computational complexity becomes linear (instead of quadratic) in the number of particles. Commonly adopted neighbor list algorithms are:

- **Verlet lists** [167, 218] : One can construct a list of neighbors based on inter-particle distances. For each particle, selected neighbors are particles within a certain threshold R, usually larger than r_c. Therefore, the neighbor lists can be reused several times, even if particles move.
- **Linked-cell lists** [52, 90]: Here, the system is divided into sub-cells bigger than the cutoff radius. For a given particle located in a cell denoted by C, the interactions are only computed with particles which belong to cells surrounding C.

In practice, the most efficient approach combines both strategies [167]. Basically, the Verlet list of a given particle is built by fetching through a larger linked-cell list previously constructed. Neighbor list algorithms are also used in the computation of long-range interactions. In fact, in most cases, long-range interactions are broken down into short and long contributions (*see* chapters 3 and 4).

1.3.4 Approximation schemes

Several interaction potentials require the evaluation of computationally time-consuming functions. As example, we can point out the Lennard-Jones potential which involves high-order power functions and dihedral angle interactions which require trigonometric functions. Fortunately, a direct evaluation of these functions can be avoided by interpolating previously tabulated values. Therefore, lookup tables are widely used for molecular simulations [97, 154, 167]. Basically, a lookup table holds information on the desired potential (and its derivatives[13]) indexed by the inter-particle distance [127].

While retrieving an interpolated value from the table, a certain degree of accuracy is lost. The accuracy can be improved by increasing either the order of interpolation or

[13] It might be useful for forces.

the node density [14] of the lookup table. In practice, lookup tables are defined up to a certain cutoff distance. As a result, the lookup table approach has been proven efficient for short-range potentials since they can be truncated [15].

Multiple-time-stepping techniques [41, 79, 200, 216] form another class of approximation schemes. These methods employ various timescales for different parts of the problem. Precisely, interactions that fluctuates slowly, are less frequently computed. Instead of computing the interaction potential at each timestep, one may compute them every n timesteps. Between two direct computations, an extrapolated interaction is used instead.

1.3.5 Fast electrostatics

Truncations schemes and lookup tables are poorly successful for long-range interactions. Several attempts have been made in order to truncate long-range potentials such as electrostatics [61, 64]. In most cases, these truncations lead to unwanted artifacts in both the particle dynamics and statistics [161].

Nevertheless, sophisticated algorithms have been developed in order to overcome the quadratic scaling of long-ranged pairwise interactions such as electrostatics [10].

- Fourier-based methods have been popularized throughout the 20^{th} century.
  - **Ewald summation** (1921) reduces the scaling of the long-range computations by splitting the electrostatic interactions (equation (1.24)) into real- and Fourier- space contributions. The real space terms are handled as short-range potentials while the Fourier space (also called k-space or reciprocal space) contributions truncate the interactions in Fourier space. Ewald summation can achieve a given accuracy in $\mathcal{O}(N^{3/2})$ operations.
  - In a similar spirit, the $\mathcal{O}(N\log(N))$ mesh solvers in Fourier space such as Particle Particle Particle Mesh (P3M) [54, 90] or Smoothed Particle Mesh Ewald (SPME) [59] split the electrostatic potential into a real- and a Fourier-part, but the Fourier part is computed on a mesh. These methods which are often called (FFT-)**Particle mesh methods**, and are the most popular approaches to the calculation of the electrostatic potential or forces in periodic systems.

    In 1980s, Hockney and Eastwood [90] have proposed P3M for the evaluation of long-range interactions in astrophysics. Here, the cost of k-space contributions

[14] number of table nodes per distance unit
[15] via a tail-correction (*e.g.* equation (1.21) for LJ potential)

(in Ewald summation) is diminished through grid-based calculations. Precisely, k-space terms are handled with some FFTs. P3M and its variants[16] [16, 54, 59] have been adopted as the method of choice for the computations of electrostatics in molecular simulations. This class of methods allows us to balance between accuracy and speed in the calculation of electrostatic interactions while reducing the computational complexity from quadratic ($\mathcal{O}(N^2)$) to quasilinear ($\mathcal{O}(N\log(N))$).

Although these FFT-based methods are often the less expensive solution for computing electrostatics on a single core, their performance on massively parallel architectures are limited by the all-to-all communications required by the FFTs [17, 46]. Therefore, some authors to consider alternatives to these methods.

- **Hierarchical approaches** like Fast Multipole Method (FMM) [77] are well known techniques for open systems. However these $\mathcal{O}(N)$ methods can be extended in periodic systems [170, 187]. Here, electrostatic interactions are treat exactly for particles within the same or neighboring sub-cells of the simulation domain. Conversely, the interactions arising from more distant particles are evaluated through multipolar expansions [46]. FMM requires a very high order to obtain the smooth forces need for molecular dynamics. This increases its cost for MD simulations, especially with periodic boundary conditions. The question about the efficiency of FMM is still open [46] since it strongly depends on the implementation. However, most studies have reported that hierarchical methods are significantly less efficient than FFT-based methods [10, 170]. From a practical point of view, hierarchical methods are far from easy to implement (efficiently), especially on multicore architectures [3, 39, 49, 50, 228].

- Several alternatives rely on iterative or **multigrid-based methods** [2, 38, 85, 181, 194, 205, 206]. Multigrid approaches utilize multiple grid levels with different spatial resolutions to compute the electrostatic interactions with a $\mathcal{O}(N)$ cost [33]. By doing so, these methods eliminate all different frequency components of the error in Fourier space [46]. Despite their linear scaling, these mesh solvers in cartesian space are slower than Fourier-based ones[10]. The Multilevel Summation

[16] FFT-Particle mesh methods may differ in the number of required FFTs and how the problem is split. In addition, the evaluation of electrostatic forces may vary from one method to another.

Method (MSM) and the Meshed Continuum Method (MCM) are two of the most decent multigrid methods.

- ◎ MSM directly calculates the electrostatic potential on a grid by using several length scales. The electrostatic potential is evaluated by splitting the so-called interaction kernel $k : (\boldsymbol{r}, \boldsymbol{r}') \mapsto ||\boldsymbol{r} - \boldsymbol{r}'||^{-1}$ into the sum of a short-range part k_0 smoothly truncated at cutoff distance r_c and slowly varying part. The slowly varying part is then restricted to a coarser grid where it is again separated into a short-range (k_1 truncated at cutoff distance $2r_c$) and a long-range contribution. This process is applied recursively up to a given coarsest grid level. Calculations are thus spread over a hierarchy of grids, and the potential computed at coarse levels is successively corrected by contributions from finer levels up to the finest grid, which yields the final potential [85, 86, 151].
- ◎ MCM uses densities with a compact support to sample the particles onto a grid and calculates an approximated electrostatic potential by solving a Poisson equation in a multigrid fashion [25, 206]. This potential is then interpolated at the particle positions. To retrieve the electrostatic potential for each particle, a short-range correction is computed based on neighboring particles. This step can take advantage of existing algorithms dedicated to short-range potentials, such as linked cells or Verlet lists.

- ▶ One may find other methods that handle electrostatics. However, they are less efficient than the methods previously described. Among others, **Local methods** [160] treat electrostatics as the static limit of Maxwellian electrodynamics, where the magnetic field and the current produced by moving charges have vanished. This approach leads to hyperbolic differential equations which can be solved with only *local* operations[17] [10]. Likewise, **Wolf summation** [62, 225] spherically truncates electrostatics while being a viable alternative to Ewald summation.

Algorithms developed in this manuscript rely on Fourier-based techniques as well as the meshed continuum method. The choice of these methods is motivated by their computational efficiency and their relative simplicity of implementation. In the previous sections, we have introduced several techniques that attempt to decrease the cost of MD simulations by speeding up the calculations of the interaction potentials. Nevertheless, the necessity to reduce the cost is still relevant for several reasons. Foremost, we want to

[17] In contrast to Elliptic differential equations (*e.g.* Poisson equation) which require *global* operations.

carry out longer simulations in order to retrieve better statistics. Likewise, an expensive all-atom representation will allow us to get more realistic states of the system. In order to hasten molecular simulations, the way in which the states of the system are generated may also be altered.

1.4 Adaptively Restrained Particle Simulation

In traditional molecular simulations, all particles of the system are moved at each timestep with respect to the hamiltonian system equation (1.1) (or equation (1.8)). It is far from obvious to claim that we need to move all particles at once in order to compute a given property, especially if this property is only related to a tiny subset of the system (*e.g.* a polymer in a solvent). To not move all particles may have a huge advantage since most interaction potentials depend explicitly/implicitly on particles' coordinates. The reader can imagine that a bond-length interaction arising from a bond formed by two particles which do not update their positions, is preserved. if such an interaction is stored, it will not require any re-computation. This idea can be extended with ease to other bonded interactions(*e.g.* when all particles that are involved in the interaction do not move). An extension to non-bonded interactions is possible but less trivial [192].

The approach we have just introduced is an example of incremental algorithms. More in detail, **incremental algorithms** allow in certain cases to reduce the cost per timestep by reusing some part of the information from the previous step [13]. Several incremental algorithms have been successfully applied to molecular simulations [174, 175, 190, 191].

In the upcoming section, we will show that incremental methods might be enabled through modifications overall dynamics of particles. In particular, ARMD relies on a modification of the system's Hamiltonian to freeze and unfreeze particles temporarily during simulations. Therefore one can avoid the need of updating interactions between frozen particles. When sufficiently many particles are restrained, the method allows significant speedups while still producing useful dynamics and statistics.

Furthermore, we will show that the corresponding modification of the generated configurations will preserve some macroscopic properties.

1.4.1 Adaptively Restrained Molecular Dynamics

In order to speed up calculations of the intermolecular potentials, ARMD was recently introduced [13, 211]. In this model, particles with sufficiently large kinetic energy ($> \varepsilon^f$)

are considered **active**, and have normal dynamics, while those with a kinetic energy below a given threshold ε^r are **restrained**, and stop moving completely.

Given a system of N particles, this behavior can be explained through the hamiltonian formalism [13], where the kinetic part of the Hamiltonian is modified, namely,

$$H_{AR}(\boldsymbol{q},\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^T\Phi(\boldsymbol{q},\boldsymbol{p})\boldsymbol{p} + V(\boldsymbol{q}). \tag{1.25}$$

Here, $\boldsymbol{q}$ is a $3N$-dimensional vector of coordinates, $\boldsymbol{p}$ is a $3N$-dimensional vector of momenta and $V(\boldsymbol{q})$ the interaction potential energy. The $3N \times 3N$ block-diagonal matrix $\Phi(\boldsymbol{q},\boldsymbol{p}) = diag[\Phi_1(\boldsymbol{q_1},\boldsymbol{p_1}),\ldots,\Phi(\boldsymbol{q_N},\boldsymbol{p_N})]$ is a modified inverse inertia matrix which reads

$$\Phi_i(\boldsymbol{q_i},\boldsymbol{p_i}) = m_i^{-1}[1-\rho_i(K_i(\boldsymbol{p}_i))]\boldsymbol{I}_{3\times 3} \tag{1.26}$$

where m_i, $\boldsymbol{q}_i$ and $\boldsymbol{p}_i$ are respectively the mass, position and momentum of particle i, $K_i(p_i) = p_i^2/2m_i$ its kinetic energy and ρ_i is its restraining function. Given the thresholds ε_i^f and ε_i^r associated to the particle i, ρ_i is defined as follows:

$$\rho_i(k) = \begin{cases} 0 & \text{if } k \geq \varepsilon_i^f \\ 1 & \text{if } k \leq \varepsilon_i^r \\ s(k) & \text{if } \varepsilon_i^r < k < \varepsilon_i^f \end{cases} \tag{1.27}$$

where s is a $\mathscr{C}^2$ function that smoothly interpolates between 0 (*restrained dynamics*) and 1 (*full dynamics*). From the so-called adaptively restrained Hamiltonian, one derives the adaptive motion of the system, namely,

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \nabla_{\boldsymbol{p}} H_{AR} = \Phi(\boldsymbol{q},\boldsymbol{p})\boldsymbol{p} + \dfrac{1}{2}\boldsymbol{p}^T\nabla_{\boldsymbol{p}}\Phi(\boldsymbol{q},\boldsymbol{p})\boldsymbol{p} \\ \dfrac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla_{\boldsymbol{q}} H_{AR} = -\nabla V - \dfrac{1}{2}\boldsymbol{p}^T\nabla_{\boldsymbol{q}}\Phi(\boldsymbol{q},\boldsymbol{p})\boldsymbol{p} \end{cases} \tag{1.28}$$

With the definition (1.26), when the particle i has its kinetic energy below ε_i^r its position is frozen for a certain period of time. The particle can resume its motion when it recovers enough momentum (or kinetic energy). When its kinetic energy is between ε_i^r and ε_f^r, the particle follows a transient motion. Meanwhile, particles whose kinetic energy exceeds ε_f^r follow the motion induced by the classic Hamiltonian H. The adaptively restrained Hamiltonian can be seen as a perturbation of the classical Hamiltonian of

molecular dynamics, since :

$$H_{AR} = \underbrace{\left[\frac{1}{2}\boldsymbol{p}^T M^{-1}\boldsymbol{p} + V(\boldsymbol{q})\right]}_{=H} + \underbrace{\frac{1}{2}\boldsymbol{p}^T\left(\Phi(\boldsymbol{q},\boldsymbol{p}) - M^{-1}\right)\boldsymbol{p}}_{+L} \tag{1.29}$$

Furthermore, we can take advantage of the fact that some particles will stay at the same location for a certain period which may correspond to several timesteps of integration. Therefore, one may skip the computation of interactions (and forces) between restrained particles since intermolecular potentials typically depend upon relative particles' positions . Incremental methods are therefore welcome to accelerate the calculation of the interaction potential as well as its derivatives. Short-range particle-particle pair interactions such as Lennard-Jones potential, were handled in [190]. By introducing a suitable neighbor list algorithm for active particles, they achieve a significant speed-up, depending on the chosen thresholds.

1.4.2 Adaptively Restrained Langevin Dynamics

Similarly to equation (1.1), the Hamiltonian proposed in section 1.4.1 covers the microcanonical ensemble (NVE). In order to generate canonical configurations, one may modify the Langevin dynamics introduced in equation (1.8) [13, 211] as follows

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} &= \nabla_{\boldsymbol{p}} H_{AR} \\ \dfrac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} &= -\nabla V - \gamma \nabla_{\boldsymbol{p}} H_{AR} + \sigma \dfrac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} \end{cases} \tag{1.30}$$

Despite the modification in the dynamics, one can show that several averaged observables remained untouched. Let us compare the statistics from both the classical and the adaptively restrained Langevin Dynamics.

Given an observable A, one can express the average over the canonical ensemble scanned by the traditional Langevin Dynamics (equation (1.10)), namely,

$$\begin{aligned} \langle A \rangle_H &= \frac{\int A(\boldsymbol{q},\boldsymbol{p}) e^{-\beta H(\boldsymbol{q},\boldsymbol{p})} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}}{\int e^{-\beta H(\boldsymbol{q},\boldsymbol{p})} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}} \\ &= \frac{\int A(\boldsymbol{q},\boldsymbol{p}) e^{-\beta [H_{AR}(\boldsymbol{q},\boldsymbol{p}) - L(\boldsymbol{q},\boldsymbol{p})]} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}}{\int e^{-\beta [H_{AR}(\boldsymbol{q},\boldsymbol{p}) - L(\boldsymbol{q},\boldsymbol{p})]} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}} \\ &= \frac{\int A(\boldsymbol{q},\boldsymbol{p}) e^{\beta L(\boldsymbol{q},\boldsymbol{p})} e^{-\beta H_{AR}(\boldsymbol{q},\boldsymbol{p})} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}}{\int e^{\beta L(\boldsymbol{q},\boldsymbol{p})} e^{-\beta H_{AR}(\boldsymbol{q},\boldsymbol{p})} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}} \end{aligned} \tag{1.31}$$

Which turns into

$$\langle A \rangle_H = \frac{\left\langle A e^{\beta L(\boldsymbol{q},\boldsymbol{p})} \right\rangle_{H_{AR}}}{\left\langle e^{\beta L(\boldsymbol{q},\boldsymbol{p})} \right\rangle_{H_{AR}}}. \tag{1.32}$$

In other terms, we can recover the suitable statistics from configurations generated according to the adaptively restrained Hamiltonian by weighting them properly.

Remarkably, the recover becomes straightforward if we consider observables that only depend on particles positions ($\boldsymbol{q}$) [13]. From the definitions in equations (1.26) and (1.29), both Φ and L solely depend upon the momenta of the particles. While using configurations generated by the adaptive dynamics, the average of a position-dependent observable A reads:

$$\begin{aligned} \langle A \rangle_{H_{AR}} &= \frac{\int A(\boldsymbol{q}) e^{-\beta H_{AR}(\boldsymbol{q},\boldsymbol{p})} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}}{\int e^{-\beta H_{AR}(\boldsymbol{q},\boldsymbol{p})} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}} \\ &= \frac{\int A(\boldsymbol{q}) e^{-\beta [H(\boldsymbol{q},\boldsymbol{p}) + L(\boldsymbol{p})]} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}}{\int e^{-\beta [H(\boldsymbol{q},\boldsymbol{p}) + L(\boldsymbol{p})]} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}} \end{aligned} \tag{1.33}$$

Meanwhile, (1.7) $\implies H(\boldsymbol{q},\boldsymbol{p}) + L(\boldsymbol{p}) = V(\boldsymbol{q}) + K(\boldsymbol{p}) + L(\boldsymbol{p})$. Therefore,

$$\begin{aligned} \langle A \rangle_{H_{AR}} &= \frac{\left[\int A(\boldsymbol{q}) e^{-\beta V(\boldsymbol{q})} \mathrm{d}\boldsymbol{q}\right] \left[\int e^{-\beta [K(\boldsymbol{p}) + L(\boldsymbol{p})]} \mathrm{d}\boldsymbol{p}\right]}{\left[\int e^{-\beta V(\boldsymbol{q})} \mathrm{d}\boldsymbol{q}\right] \left[\int e^{-\beta [K(\boldsymbol{p}) + L(\boldsymbol{p})]} \mathrm{d}\boldsymbol{p}\right]} \\ &= \frac{\int A(\boldsymbol{q}) e^{-\beta V(\boldsymbol{q})} \mathrm{d}\boldsymbol{q}}{\int e^{-\beta V(\boldsymbol{q})} \mathrm{d}\boldsymbol{q}} \end{aligned}$$

Finally, one can multiply both the numerator and the denominator by $\int e^{-\beta K(\boldsymbol{p})} \mathrm{d}\boldsymbol{p}$, hence

$$\begin{aligned} \langle A \rangle_{H_{AR}} &= \frac{\int A(\boldsymbol{q}) e^{-\beta [V(\boldsymbol{q}) + K(\boldsymbol{p})]} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}}{\int e^{-\beta [V(\boldsymbol{q}) + K(\boldsymbol{p})]} \mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p}} \\ \langle A \rangle_{H_{AR}} &= \langle A \rangle_H \end{aligned} \tag{1.34}$$

This result is extremely powerful for several properties such as the radius of gyration of a polymer, the radial distribution of a system.

1.4.3 Restraining parameters

Let us consider a system of N particles simulated with ARMD. The adaptively restrained dynamics is entirely controlled by the choice of the restraining parameters $\left(\varepsilon_i^r, \varepsilon_i^f\right)_{i \in \{1,\dots,N\}}$. For the sake simplicity, let us assume that $\varepsilon_i^f = \varepsilon^f$ and $\varepsilon_i^r = \varepsilon^r$, $\forall i \in \{1,\dots,N\}$. In practice, parameters are not defined individually for each particle. Instead, several particles

share the same parameterization. For example, in a solute-solvent simulation, the same configuration can be given for solvent particles.

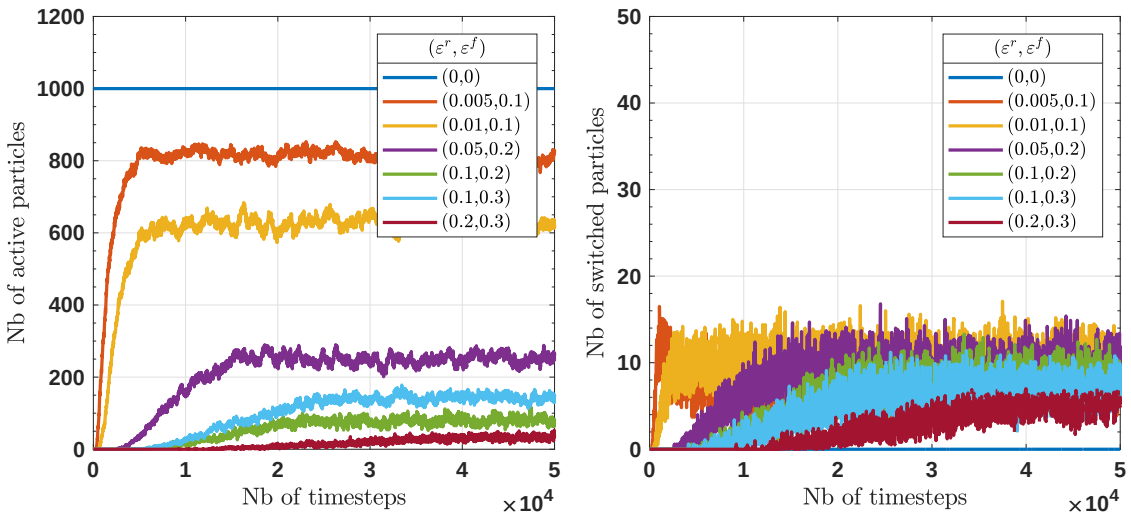
Furthermore, the choice of restraining parameters must be done carefully. Foremost, ε^r affects the proportion of restrained particles which may fluctuate throughout the simulation. In particular, small (*resp.* large) values of ε^r often increase (*resp.* decrease) the average number of restrained particles. Incremental algorithms will benefit from an increased number of restrained particles. Conversely, when the number of restrained particles is too large, a modest number of uncorrelated system states can be extracted from the simulated trajectory. Thus, long simulations may be needed in order to recover suitable statistics.

On the other hand, ε^f affects the stability of the simulation. In fact, when ε^r and ε^f are too close, particles may switch too quickly from restrained to the full dynamics, leading to unstable simulations. Conversely, when ε^r and ε^f are too far from each other, particles may spend too much time in the transient dynamics. As a result, time-average properties may converge slowly [210].

In order to be more acquainted with the adaptively restrained dynamics, we analyzed the evolution of a system of $N = 1000$ particles of argon. These particles are placed in a 50 Å simulation. The inter-particle interactions are modeled with LJ potential ($\sigma = 3.405$ Å and $\epsilon = 0.2381$ kcal/mol [142]). The dynamics were governed by adaptive restrained Langevin dynamics ($T = 300$K). The equations of motions were integrated following [12, 210] with an integration timestep set to 1 fs. We monitor the effect of restraining parameters on the number of restrained particles (N_r) and the number of active particles (N_a). In particular, the time evolution of N_a is reported in figure 1.8(1). Since N_a fluctuates throughout the simulation, we also output N_s which corresponds to the number of particles that switch either from active to restrained or from restrained to active (figure 1.8(2)). Interestingly, N_s is often marginal with respect to N_a and N_r. Finally, the mean values of N_a, N_r and N_s are shown in table 1.2.

$(\varepsilon^r, \varepsilon^f)$	$\langle N_r \rangle$	$\langle N_a \rangle$	$\langle N_s \rangle$	$\langle N_s \rangle / \langle N_r \rangle$
(0.000, 0.000)	0	1000	0	–
(0.005, 0.100)	213	787	7.40	0.035
(0.010, 0.100)	405	595	10.05	0.025
(0.050, 0.200)	793	207	7.87	0.010
(0.100, 0.200)	943	56	5.59	0.006
(0.100, 0.300)	902	98	4.96	0.005
(0.200, 0.300)	985	15	2.74	0.003

Table 1.2: Average number of active/restrained/switched particles throughout 50 ps adaptively restrained simulations of Argon.



(1) Number of active particles

(2) Number of switched particles

Figure 1.8: Time evolution of the number of active/switched particles through a 50 ps adaptively restrained simulations of Argon.

1.5 Contributions

In order to take advantage of adaptive restraints, simulations must use incremental force update algorithms that skip the calculation of forces involving restrained particles. ARMD allows incremental computations which can ultimately reduce the computational cost of the simulation. Although such algorithms have been demonstrated for short-range interactions [192], no method had yet been proposed for long-range interactions.

The primary goal of this thesis was therefore to contribute to the acceleration of molecular simulations by considering incremental algorithms which may reduce the computational cost of long-range interactions. The main contributions of this thesis can be summarized as follows:

- In chapter 3, we propose an improvement of various Fourier-based methods in order to efficiently compute long-range interactions, especially electrostatics. Foremost, Ewald summation was successfully sped up by accelerating both real space and k-space calculations. In addition, mesh solvers in Fourier space such as Particle Particle Particle Mesh (P3M) can directly benefit of the real space's acceleration.

  Moreover, we rethink the treatment of k-space computations on a grid. Basically, we utilize a multi-level approach by decomposing the problem on two main contributions which are solved with different grid resolutions. In addition, the two sub-problems are solved asynchronously (on different timescales). We show that this strategy can maintain the desired accuracy while reducing the cost of the k-space calculations. However, the ideal performance of the proposed approach is limited by the proportion of particles that switches their state.

- In chapter 4, we propose a multigrid-based approach which incrementally handles long-range interactions. Although multigrid methods are natively more time-consuming than Fourier-based techniques, they offer a linear scaling with respect to the number of particles. We show that the proposed approach can be effectively applied to systems where only few particles update their positions at each timestep. We are able to outperform traditional Fourier-based solvers on several applications with ARMD. This work has been published in the Journal of Computational Chemistry (J. Comput. Chem.) [57].

- In chapter 5, we apply ARMD for the study of translocations of a polyelectrolyte. The threading of a charged polymer through a nanopore is of interest in several fields such as DNA sequencing. However, most of studies limit themselves to

simulations with implicit water. These simulations are less expensive in terms of CPU resources while neglecting hydrodynamic effects. We show that one can reduce the cost of explicit water simulations of the translocation by adaptively restraining the motion of the solvent particles. Thanks to incremental algorithms, the proposed approach decreases the cost of electrostatic computations.

CHAPTER 2

MATHEMATICAL TOOLBOX

Contents

2.1 Periodic images

Under p.b.c, the simulation box is represented as an infinitely replicated supercell. The supercell is described by three vectors $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$. Let us consider that a particle i located at $\boldsymbol{x}_i$, is enclosed in the simulation box. The periodic images of the particle i are located in the replicated cells and their coordinates read $\boldsymbol{x}_i + (n_1\boldsymbol{a}_1 + n_2\boldsymbol{a}_2 + n_3\boldsymbol{a}_3)$. Here n_1, n_2, n_3 are arbitrary integers.

For the sake of simplicity, we restrict ourselves to a cubic simulation box. As a result, we can define the characteristic length of the supercell as $L = ||a_1|| = ||a_2|| = ||a_3||$. In this case, periodic images of particle i are located at $\boldsymbol{x}_i^{\boldsymbol{n}} = \boldsymbol{x}_i + \boldsymbol{n}L$ with $\boldsymbol{n} \in \mathbb{Z}^3$.

2.2 Poisson equation and long-range interactions

Let us consider a system of N particles at location $(\boldsymbol{x}_i)_{i\in\{1,\dots,N\}}$ possessing point charges $(q_i)_{i\in\{1,\dots,N\}}$. Particles are placed in a cubic simulation box $\mathscr{B}$ of length L and p.b.c are assumed. The total electrostatic potential stored in this system energy reads

$$E = \frac{1}{2}\sum_{\boldsymbol{n}\in\mathbb{Z}^3}\sum_{i=1}^{N}\sum_{j=1}^{N}{}' \frac{q_i q_j}{4\pi\epsilon_0}\frac{1}{||\boldsymbol{x}_j^{\boldsymbol{n}} - \boldsymbol{x}_i||}. \tag{2.1}$$

We used the $'$ symbol to exclude the term $j = i$, if and only if $\boldsymbol{n} = \boldsymbol{0}$. One can rewrite the electrostatic potential energy with the help of the electrostatic potential Φ_i due to all point charges except the one at $\boldsymbol{x}_i$

$$E = \frac{1}{2}\sum_{i=1}^{N} q_i \Phi_i(\boldsymbol{x}_i) \tag{2.2}$$

$$\Phi_i(\boldsymbol{x}_i) = \sum_{\boldsymbol{n}\in\mathbb{Z}^3}\sum_{j=1}^{N}{}' \frac{1}{4\pi\epsilon_0}\frac{q_j}{||\boldsymbol{x}_j^{\boldsymbol{n}} - \boldsymbol{x}_i||} \tag{2.3}$$

An analogous formulation exists for the gravitational potential energy. More generally, the electrostatic (*resp.* gravitational) potential is linked to the distribution of charges (*resp.* masses) via a Poisson equation.

$$\forall \boldsymbol{x} \in \mathbb{R}^3, \qquad \Delta\Phi_i(\boldsymbol{x}) = -\frac{1}{\epsilon_0}\rho_i(\boldsymbol{x}) \tag{2.4}$$

where

$$\rho_i : \boldsymbol{x} \mapsto \sum_{\boldsymbol{n}\in\mathbb{Z}^3}\sum_{j=1}^{N}{}' q_j \delta(||\boldsymbol{x} - \boldsymbol{x}_j + \boldsymbol{n}L||) \tag{2.5}$$

denotes the distribution of all point charges except the charge i. Here, this distribution can be represented with the help of Dirac δ functions. Therefore, another way of calculating electrostatic energy can use the Poisson formulation. However, this formulation implies that it is necessary to solve, at each timestep, N Poisson equations in order to retrieve coulombic forces. This is computationally infeasible even for systems containing few hundreds atoms. Solving (only once) the following equation

$$\forall \boldsymbol{x} \in \mathbb{R}^3, \qquad \Delta\Phi(\boldsymbol{x}) = -\frac{1}{\epsilon_0} \sum_{\boldsymbol{n}\in\mathbb{Z}^3} \sum_{j=1}^{N} q_j \delta(||\boldsymbol{x} - \boldsymbol{x}_j + \boldsymbol{n}L||). \tag{2.6}$$

gives the potential created by all particles. For each particle i, one may retrieve the desired potential $\Phi_i(\boldsymbol{x}_i)$ by subtracting a term which corresponds to the self-interaction added to equation (2.5). For the sake of readability, we can rewrite equation (2.6) with respect to p.b.c

$$\forall \boldsymbol{x} \in \mathscr{B}, \qquad \Delta\Phi(\boldsymbol{x}) = -\frac{1}{\epsilon_0} \sum_{j=1}^{N} q_j \delta(||\boldsymbol{x} - \boldsymbol{x}_j||). \tag{2.7}$$

An analytical solution of Poisson equation is not always available. Numerical methods are often needed for the resolution of this class of partial differential equation (PDE). In practice, the Poisson equation is solved using Fourier transforms (as it is done in chapter 3) or using the multigrid method (*see* chapter 4). Furthermore, the Dirac impulses are not adequate for numerical resolutions, due to the discontinuities introduced by every particle. To get rid of this issue, Dirac impulses are often replaced by smooth charge distributions, especially[1] point symmetric densities.

Definition 2.1 (Point symmetric densities)**.** *Let us consider a real function* $\varrho_{r_c} : \mathbb{R}_+ \to \mathbb{R}_+$ *with a compact support* $r_c > 0$ *, i.e.* $\left(supp\left(\varrho_{r_c}\right) = [0, r_c]\right)$.

A function $\rho_{r_c} : \mathbb{R}^3 \to \mathbb{R}_+$ *defined by*

$$\rho_{r_c}(\boldsymbol{x}) := \varrho_{r_c}(||\boldsymbol{x}||)$$

is called a point symmetric density if

$$\int_{\mathbb{R}^3} \phi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1.$$

[1] Gaussian distributions or fast-decaying densities are good alternatives to point symmetric densities.

For the sake of simplicity, we will write point symmetric densities as follows

$$\rho_{r_c}(\boldsymbol{x}) := \rho_{r_c}(||\boldsymbol{x}||).$$

Moreover, these functions are often used in order to construct numerical approximations of long-range potentials. In fact, beyond the cutoff radius r_c, the potential induced by a point symmetric density matches the potential induced by a Dirac impulse.

Lemma 2.1 (The fundamental lemma)**.** *Let ρ_{r_c} be a point symmetric density with cutoff radius r_c. We consider the following Poisson equations*
$\forall \boldsymbol{x} \in \mathscr{B}$,

$$\Delta\Phi_a(\boldsymbol{x}) = \rho_{r_c}(||\boldsymbol{x}||)$$

$$\Delta\Phi_b(\boldsymbol{x}) = \delta(||\boldsymbol{x}||)$$

with suitable boundaries conditions[2]*. The corresponding solutions follow $\forall ||\boldsymbol{x}|| > r_c$,*

$$\Phi_a(\boldsymbol{x}) = \Phi_b(\boldsymbol{x}).$$

The proof of this classical result is straightforward and rely on a direct calculation of the solutions through a convolution with the 3D Green's function $g(||\boldsymbol{x}||) = \dfrac{1}{4\pi||\boldsymbol{x}||}$. The reader should refer to [25] for detailed calculations.

2.3 Construction of smooth charge densities

A numerical resolution of the Poisson equation requires that the right-hand side (r.h.s.) must be smooth. Therefore, we will construct point symmetric densities that fulfill this condition. In addition, some methods such as MCM require an analytic expression of the corresponding potential. Therefore, one may use charge densities that are described with the help of cardinal B-splines.

[2] If p.b.c are imposed, the equality is valid up to a real constant. If Dirichlet (*resp.* Neumann) conditions are prescribed, the equality is strict since each Poisson equation admits a unique solution.

2.3.1 Charge densities described with cardinal B-splines.

Definition 2.2 (Cardinal B-splines)**.** *A order-m cardinal B-spline, here denoted by* $B^{(m)}$*, is a* $\mathscr{C}^{m-2}[0,m]$ *function with the following properties [44, 45, 145]:*

- $supp\left(B^{(m)}\right) = [0,m]$*;*
- *at each interval* $[k,k+1]$*,* $0 \leq k \leq m-1$*;*
- $\forall t \in [0,m]$*,* $B^{(m)}(t) = B^{(m)}(m-t)$*.*

Remark 2.1. *By defining* $B_{r_c}^{(m)} : r \mapsto B^{(m)}\left(\frac{m}{2}\left(\frac{r}{r_c}+1\right)\right)$*, we can construct a centered B-spline with* $supp\left(B_{r_c}^{(m)}\right) = [-r_c, r_c]$*.*

Definition 2.3. *A charge density described with a order-m B-spline, reads*

$$\rho_{r_c}^{(m)}(\boldsymbol{r}) := \rho_{r_c}(||\boldsymbol{r}||) = \frac{1}{A} B_{r_c}^{(m)}(||\boldsymbol{r}||)$$

where $\frac{1}{A}$ *is a normalizing constant.*

Precisely, A is given by

$$\begin{aligned} A &= \int_{\mathbb{R}^3} B_{r_c}^{(m)}(||\boldsymbol{r}||)\mathrm{d}\boldsymbol{r} \\ &= \int_0^{r_c} 4\pi r^2 B_{r_c}^{(m)}(r)\mathrm{d}r. \end{aligned} \tag{2.8}$$

The charge density used by Bolten [25], in their original paper, is a centered quadratic B-spline. This density is described by a order-3 B-spline, namely,

$$\rho_{r_c}^{(3)}(\boldsymbol{r}) = \begin{cases} \dfrac{-486r^2 + 162r_c^2}{32\pi r_c^5} & \text{if } r < \dfrac{r_c}{3} \\ \dfrac{243(r-r_c)^2}{32\pi r_c^5} & \text{if } \dfrac{r_c}{3} \leq r < r_c \\ 0 & \text{if } r \geq r_c \end{cases} \tag{2.9}$$

where $r = ||\boldsymbol{r}||$. The corresponding potential can be determined analytically by convolving $\rho_{r_c}^{(3)}$ with the Green's function g.

$$\phi^{(3)}(\boldsymbol{r}) = \begin{cases} \dfrac{3(81r^4 - 90r^2r_c^2 + 65r_c^4)}{320\pi r_c^5} & \text{if } r < \dfrac{r_c}{3} \\ \dfrac{-243r^5 + 810r^4r_c - 810r^3r_c^2 + 405rr_c^4 - 2r_c^5}{640\pi r r_c^5} & \text{if } \dfrac{r_c}{3} \le r < r_c \\ \dfrac{1}{4\pi r} & \text{if } r \ge r_c. \end{cases} \tag{2.10}$$

For this study, we evaluated the use of higher order cardinal splines. For instance, a charge density described by the 5^{th} order B-spline reads

$$\rho_{r_c}^{(5)}(\boldsymbol{r}) = \begin{cases} \dfrac{54(81r^4 - 54r^2r_c^2 + 11r_c^4)}{64\pi r_c^7} & \text{if } r < \dfrac{r_c}{3} \\ \dfrac{-243r^4 + 540r^3r_c - 378r^2r_c^2 + 60rr_c + 17r_c^4}{64\pi r_c^7} & \text{if } \dfrac{r_c}{3} \le r < \dfrac{2r_c}{3} \\ \dfrac{2187(r - r_c)^4}{64\pi r_c^7} & \text{if } \dfrac{2r_c}{3} \le r < r_c \\ 0 & \text{if } r \ge r_c, \end{cases} \tag{2.11}$$

While its corresponding potential reads

$$\phi^{(5)}(\mathbf{r}) = \begin{cases} -\dfrac{7290r^6 - 10206r^4r_c^2 + 6930r^2r_c^4 - 3346r_c^6}{4480\pi r_c^7} & \text{if } r < \dfrac{r_c}{3} \\ \dfrac{32805r^7 - 102060r^6r_c + 107163r^5r_c^2 - 28350r^4r_c^3 - 16065r^3r_c^4 + 9933rr_c^6 + 10r_c^7}{13440\pi r r_c^7} & \text{if } \dfrac{r_c}{3} \le r < \dfrac{2r_c}{3} \\ -\dfrac{3645r^7 - 20412r^6r_c + 45927r^5r_c^2 - 5130r^4r_c^3 + 25515r^3r_c^4 - 5103rr_c^6 + 338r_c^7}{4480\pi r r_c^7} & \text{if } \dfrac{2r_c}{3} \le r < r_c \\ \dfrac{1}{4\pi r} & \text{if } r > r_c. \end{cases} \tag{2.12}$$

2.3.2 Charge densities described with polynomials.

Alternatively, one can construct densities with the help of high order polynomials. In particular, one can use a restriction of order-$2m$ polynomials defined as follows

$$P_{r_c}^{(2m)}(r) = \left(r^2 - r_c^2\right)^m \mathbb{1}^{[0,r_c]}(r) \tag{2.13}$$

where $\mathbb{1}^{[0,r_c]}$ indicates to the characteristic function of $[0, r_c]$. Thus,

Definition 2.4. *a smooth charge density described by $P_{r_c}^{(2m)}$ reads:*

$$\rho_{r_c}(||\boldsymbol{r}||) = \frac{1}{A} P_{r_c}^{(2m)}(||\boldsymbol{r}||).$$

Again $\dfrac{1}{A}$ defines a normalizing constant such that

$$A = \int_0^{r_c} 4\pi r^2 P_{r_c}^{(2m)}(r)\mathrm{d}r. \tag{2.14}$$

For instance, when using an order-10 polynomial, the charge density ρ reads:

$$\rho_{rc}(r) = \frac{9009}{256\pi r_c^{13}}\left(r_c^2 - r^2\right)^5 \qquad \text{if } r \le r_c. \tag{2.15}$$

Thus, the corresponding potential reads

$$\phi(r) = \frac{a_{12}r^{12} - a_{10}r^{10}r_c^2 + a_8 r^8 r_c^4 + a_6 r^6 r_c^6 + a_4 r^4 r_c^8 + a_2 r^2 r_c^{10} + a_0 r c^{12}}{8192\pi r_c^{13}} \tag{2.16}$$

$$\text{where} \qquad a_{12} = 462, \quad a_{10} = 3276, \quad a_8 = 1001,$$
$$a_6 = -1716, \quad a_4 = 18018, \quad a_2 = -12012, \quad a_0 = 6006.$$

Figure 2.1 shows several densities described by B-Splines (order 3,5,7) or (order 6, 10, 14) polynomials.

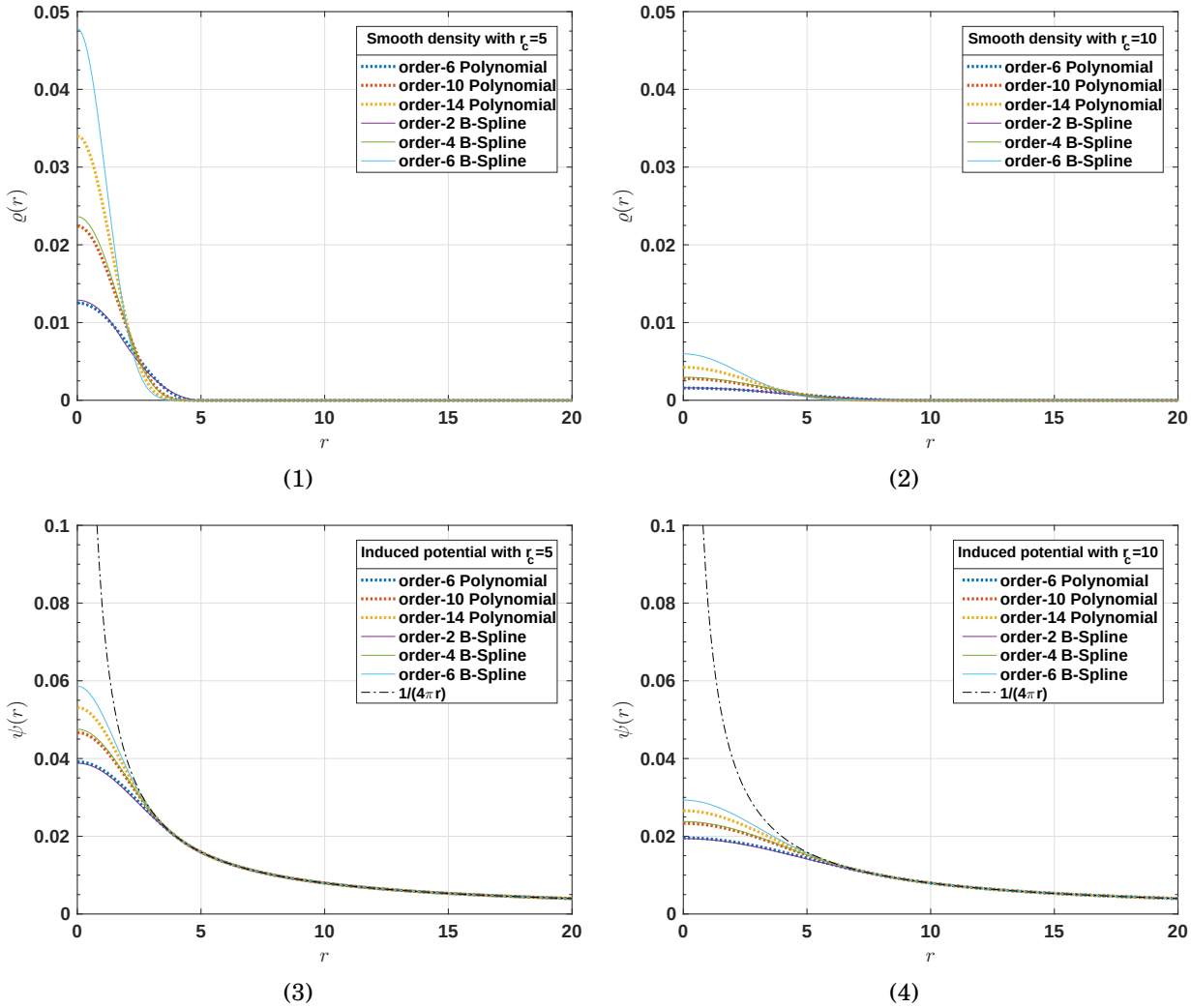


Figure 2.1: Comparison of various smooth charge densities (Top) with $r_c = 5$ (left) and $r_c = 10$ (right). B-Spline densities (order 3,5,7) are shown with solid line. 4^{th}, 10^{th} and 14^{th} order polynomials are represented with dots. For each configuration, the corresponding potential is plotted at the bottom. The desired potential $r \mapsto \frac{1}{4\pi r}$ (black dashed dots) is shown as reference.

2.4 Fourier Transforms

Fourier transforms are a powerful tool for the resolution of the Poisson equation. They are intensively used for the treatment of electrostatics (*see* chapter 3). We recommend the book by Bracewell [27] or by Brigham [34] for a complete overview of Fourier Transforms and their applications. Here, we will restrict ourselves to a few definitions.

2.4.1 Continuous Fourier Transforms

Definition 2.5 (Forward Fourier Transform)**.** *Given the aforementioned simulation box under p.b.c, we define the Fourier transform of a function $f \in L^1(\mathscr{B})$ as*

$$\tilde{f}(\boldsymbol{k}) = \int_{\mathscr{B}} f(\boldsymbol{r}) e^{-i\boldsymbol{k}.\boldsymbol{r}} \mathrm{d}\boldsymbol{r} \tag{2.17}$$

where $\boldsymbol{k} \in \{2\pi \boldsymbol{n}/L : \boldsymbol{n} \in \mathbb{Z}^3\}$ denotes a reciprocal or $\boldsymbol{k}$-space vector.

Fourier transforms are often employed for solving Poisson equations. In fact, the following Poisson equation

$$\Delta\Phi = -\frac{\varphi}{\epsilon_r} \tag{2.18}$$

gives in reciprocal space

$$k^2\tilde{\Phi} = -\frac{\tilde{\varphi}}{\epsilon_r}. \tag{2.19}$$

Therefore, a straightforward expression of the potential field is available in reciprocal space. Fortunately, the Fourier transform can be reversed.

Definition 2.6 (Backward Fourier Transform)**.** *Without loss of generality, the backward (or inverse) Fourier transform is defined as*

$$f(\boldsymbol{r}) = \frac{1}{V} \sum_{\boldsymbol{k}} \tilde{f}(\boldsymbol{k}) e^{i\boldsymbol{k}.\boldsymbol{r}} \tag{2.20}$$

where $V = L^3$ is the volume[3] of $\mathscr{B}$.

2.4.2 Discrete Fourier Transform

In signal processing, the discrete Fourier Transform (DFT) is the equivalent of the continuous Fourier Transform for signals known only for a finite sequence of data. Moreover, DFTs are often employed to solve Poisson problem on grids.

[3] Here, we use a three dimensional cubic domain.

Definition 2.7 (Forward Discrete Fourier Transform)**.** *Let us consider a d-dimensional array* $(f_{n_1,\dots,n_d})$ *of length* $M = N_1 N_2 \dots N_d$*. Its DFT reads*

$$\hat{f}_{k_1,\dots,k_d} = \sum_{n_1=0}^{N_1-1} \left[\omega_{N_1}^{k_1 n_1} \sum_{n_2=0}^{N_2-1} \left[\omega_{N_2}^{k_2 n_2} \cdots \sum_{n_d=0}^{N_d-1} \omega_{N_d}^{k_d n_d} f_{n_1,\dots,n_d} \right] \right] \tag{2.21}$$

where $\omega_{N_l} = exp(-2\pi i/N_l)$ *and the output indices run from* $k_l = 0, 1 \dots, N_l - 1$..

Remark 2.2. *For the sake of readability, we may use the following notation:*

$$\hat{f}_{\boldsymbol{k}} = \sum_{\boldsymbol{n}=\boldsymbol{0}}^{\boldsymbol{N-1}} f_{\boldsymbol{n}} \exp(-i\boldsymbol{k}.\boldsymbol{n}). \tag{2.22}$$

where
$\boldsymbol{n} = (n_1, n_2, \dots, n_d)$, $\boldsymbol{k} = \left(\frac{2\pi}{N_1} k_1, \frac{2\pi}{N_2} k_2, \dots, \frac{2\pi}{N_d} k_d \right)$ *and* $\boldsymbol{N-1} = (N_1 - 1, \dots, N_d - 1)$.

Definition 2.8 (Backward Discrete Fourier Transform)**.** *Using the notation introduced in remark 2.2, we define the inverse of the d-dimensional DFT as*

$$f_{\boldsymbol{n}} = \left(\prod_{l=1}^{d} N_l \right)^{-1} \sum_{\boldsymbol{k}} \hat{f}_{\boldsymbol{k}} \exp(i\boldsymbol{k}.\boldsymbol{n}). \tag{2.23}$$

2.4.3 Fast Fourier Transform

Moreover, the DFT can be seen as a matrix vector product. Therefore, a naive evaluation leads to $\mathcal{O}(M^2)$ operations when the length of the input sequence is M. Fortunately, the efficiency of the evaluation can be improved by FFT which reduces the computational complexity to $\mathcal{O}(M \log(M))$. There are numerous FFT algorithms and their efficiency often depends on the factorization of M. For instance, the method proposed by Cooley and Tukey [47] is extremely successful when M is composite[4,5]. When M is a prime number, [24, 172] are more efficient. In most cases, FFT algorithms enhance some performance by precomputing and saving the trigonometric constant coefficients $\omega_{N_l} = exp(-2\pi i/N_l)$. These constants are called twiddle factors. Their cost of evaluation is too high for on-the-fly calculations.

From a practical point of view, one resorts to software libraries dedicated to the computation of discrete Fourier transforms. Although, its performance may depend on the hardware configuration, the Fastest Fourier Transform in the West [68] is a

[4] A composite number is a positive integer that can be formed by multiplying together two smaller positive integers.

[5] The best performance is achieved when each N_l is composite.

widely used library. In figure 2.2, we output the average cost in wall clock time of a 3-dimensional FFT performed with this library. The length of the input sequence is $M = N \times N \times N$. While increasing the length of the sequence, the CPU time increases as $\mathcal{O}(N^3 \log(N))$, albeit a highly fluctuating evolution. Interestingly, the evolution is more smooth for composite values of N. Thus, we will preferentially restrict to those sizes.

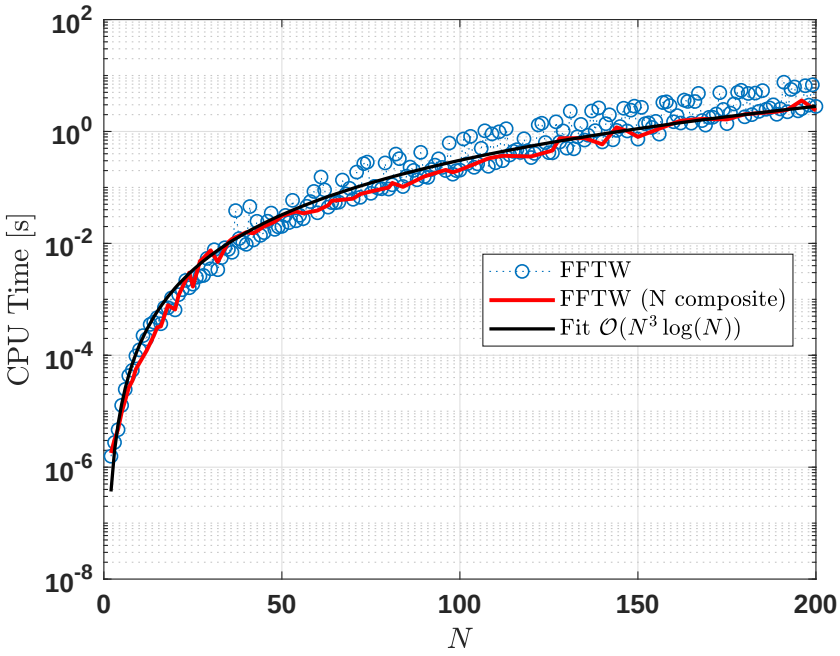


Figure 2.2: FFTW performance on various input sizes.

2.4.4 Some examples of Fourier Transforms

Here, we give some examples of Fourier transform of specialized distributions.

The distribution of point charges – The Fourier transform of the distribution of all point charges is given by

$$\varphi(\boldsymbol{x}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^3} \sum_{j=1}^{N} q_j x \delta(||\boldsymbol{r} - \boldsymbol{r}_j + \boldsymbol{n}L||) \Longrightarrow \tilde{\varphi}(\boldsymbol{k}) = \sum_{j=1}^{N} q_j e^{-i\boldsymbol{k}.\boldsymbol{r}_j} \tag{2.24}$$

The distribution of Gaussian charges – Let consider a Gaussian distribution G defined as follows

$$G_\sigma(\boldsymbol{r}) = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{r^2}{2\sigma^2}\right). \tag{2.25}$$

Gaussians are a good approximation of Dirac δ functions since

$$\lim_{\sigma \to 0} G_\sigma(\boldsymbol{x}) = \delta(\boldsymbol{x}) \tag{2.26}$$

Let us replace point charges located at $(\boldsymbol{x}_i)$ by Gaussian charges namely,

$$\varphi(\boldsymbol{x}) = \sum_{\boldsymbol{n} \in \mathbb{Z}^3} \sum_{j=1}^{N} q_j G_\sigma(||\boldsymbol{x} - \boldsymbol{x}_j + \boldsymbol{n}L||). \tag{2.27}$$

The corresponding Fourier transform reads

$$\begin{aligned}
\tilde{\rho}(\boldsymbol{k}) &= \int_{\mathscr{B}} \sum_{\boldsymbol{n} \in \mathbb{Z}^3} \sum_{j=1}^{N} q_i G_\sigma(||\boldsymbol{x} - \boldsymbol{x}_j + \boldsymbol{n}L||) exp(-i\boldsymbol{k}.\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \sum_{j=1}^{N} \sum_{\boldsymbol{n} \in \mathbb{Z}^3} \int_{\mathscr{B}} q_i G_\sigma(||\boldsymbol{x} - \boldsymbol{x}_j + \boldsymbol{n}L||) exp(-i\boldsymbol{k}.\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \sum_{j=1}^{N} \sum_{\boldsymbol{n} \in \mathbb{Z}^3} \int_{\mathscr{B}_{\mathbf{n}}} q_i G_\sigma(||\boldsymbol{r} - \boldsymbol{x}_j||) exp(-i\boldsymbol{k}.\boldsymbol{r} - i\boldsymbol{k}.\boldsymbol{n}L) \mathrm{d}\boldsymbol{r} \\
&\qquad \text{With } \mathscr{B}_{\boldsymbol{n}} := \mathscr{B} + \boldsymbol{n}L. \\
&= \sum_{j=1}^{N} \sum_{\boldsymbol{n} \in \mathbb{Z}^3} \int_{\mathscr{B}_{\mathbf{n}}} q_i G_\sigma(||\boldsymbol{r} - \boldsymbol{x}_j||) exp(-i\boldsymbol{k}.\boldsymbol{r}) \mathrm{d}\boldsymbol{r} \\
&\qquad \text{Since } exp(-i\boldsymbol{k}.\boldsymbol{n}L) = 1. \\
&= \sum_{j=1}^{N} \int_{\mathbb{R}^3} q_i G_\sigma(||\boldsymbol{r} - \boldsymbol{x}_j||) exp(-i\boldsymbol{k}.\boldsymbol{r}) \mathrm{d}\boldsymbol{r}
\end{aligned} \tag{2.28}$$

In the above derivation, we have used the fact that $(\mathscr{B}_{\boldsymbol{n}})_{\boldsymbol{n} \in \mathbb{Z}^3}$is a partition of $\mathbb{R}^3$. Finally,

$$\tilde{\varphi}(\boldsymbol{k}) = \sum_{j=1}^{N} q_j e^{-i\boldsymbol{k}.\boldsymbol{x}_j} e^{-\sigma^2 k^2/2}. \tag{2.29}$$

2.5 Multigrid

Multigrid method is a long-established iterative solver for linear systems of equations

$$A\boldsymbol{u} = \boldsymbol{f} \tag{2.30}$$

where $A \in \mathbb{R}^{N\times N}$ and $\boldsymbol{u}, \boldsymbol{f} \in \mathbb{R}^N$.

This approach was intensively studied and popularized by Brandt [29, 30] and Hackbusch [82]. However, for good introductions and a complete overview of multigrid methods, we, for instance, refer to [33, 209, 224]. In addition, the reader could find details on effective multigrid implementation in [76, 96, 114, 201]. For almost 40 years, this field of research has flourished and many researchers have contributed to it. The *MGNet Bibliography*[6] (more than 3500 entries) is a rich repository of information and articles on multigrid method.

It should be noted that most MD simulations often employ simple geometries. For instance, in both NVE and NVT simulations, the studied domain is represented as a box which can be orthogonal or triclinic. In addition, this simulation box does not change throughout. Since ARMD has been successfully studied for microcanonical and canonical simulations (an extension to isothermal–isobaric ensemble case was not presented yet), we adopt geometric multigrid method. This approach is indeed compelling for problems arising from PDE with simple geometries. Conversely, an algebraic multigrid method may be more effective for more complex geometries [178].

Since chapter 4 depicts a multigrid-based method, this section will give a brief introduction to the main ideas of the geometric full multigrid method. Thus, let us consider the following 3D Poisson problem :

Find $u : \Omega \mapsto \mathbb{R}$ such that

$$-\Delta u(x,y,z) = f(x,y,z), \qquad (x,y,z) \in \Omega \tag{2.31}$$

where Ω is a continuous convex domain in 3D space with suitable conditions on its boundary $\delta\Omega$. Here, we assume periodic boundary conditions.

[6] http://www.mgnet.org/mgnet-bib.html

2.5.1 Discretization

Given a Cartesian grid Ω^h of constant mesh size with a formal discretization parameter h, the terms in equation (2.31) are discretized. Equation (2.31) becomes a linear system of equations,

$$A_h u_h = f_h. \tag{2.32}$$

A discretization of the Laplace operator in equation (2.31) requires a specific number of neighboring grid points with appropriate coefficients used for evaluation, giving rise to a certain discretization order [10]. The 7-point stencil central difference scheme is the simplest discretization, namely,

$$\begin{aligned} 6u_{i,j,k} - (u_{i+1,j,k} + u_{i-1,j,k} + u_{i,j+1,k} \\ + u_{i,j-1,k} + u_{i,j,k+1} + u_{i,j,k-1}) \quad &= \quad h^2 f_{i,j,k}. \end{aligned} \tag{2.33}$$

where i, j, k are coordinates of a specific node of the cartesian grid.

Nevertheless, for reasons of locality in implementations with emphasis on strong parallel scalability, a compact 27-point stencil of fourth order is typically used (*see* [8]).

2.5.2 Geometric multigrid

Multigrid can be easily motivated by taking an in-depth look at simple iterative relaxation schemes such as as Gauss Seidel, Jacobi or successive over-relaxation [179]. Using one of the aforementioned methods, one determines the approximate solution after p iterations $\boldsymbol{u}_h^{(p)}$ of the equation (2.32) as

$$\boldsymbol{u}_h^{(p)} = \boldsymbol{u}_h^{(p-1)} + B_h(\boldsymbol{f} - A_h \boldsymbol{u}_h^{(p-1)}). \tag{2.34}$$

Here B_h denotes an approximate inverse. For instance, Jacobi method uses $B_h = D_h^{-1}$ where D_h is the diagonal component of A. Relaxation methods minimize the algebraic error $\boldsymbol{e}_h = \boldsymbol{u}_h - \boldsymbol{u}_h^{(p)}$. However, since the solution $\boldsymbol{u}_h$ is unknown, this error is knotty to determine. Nevertheless, one can estimate the quality of the approximation $\boldsymbol{u}_h^{(p)}$ by computing the residual (or defect):

$$\boldsymbol{r}_h := \boldsymbol{f}_h - A\boldsymbol{u}_h^{(p)} = \boldsymbol{f}_h - A_h(\boldsymbol{e}_h - \boldsymbol{u}_h) = A_h \boldsymbol{e}_h. \tag{2.35}$$

Here, the defect $\boldsymbol{r}_h$ vanishes if the error is zero. A spectral analysis of the error components reveals that conventional iterative solvers satisfactorily smooth the high-frequency parts of the error whereas the convergence of the low-frequency parts is very poor [33, 179].

Furthermore, in case of compact discretizations, the iteration scheme converges the more slowly, the finer the grid is resolved [10]. Fortunately, as the produced error is shown «*smooth*», multigrid methods switch to a coarser grid, on which «*smooth*» becomes «*rough*» and low frequencies act like higher frequencies. The main idea of the multigrid algorithm is the so-called **coarse grid correction**:

1. Determine an approximation $\boldsymbol{u}$ of $\boldsymbol{u}_h$ on the original grid Ω^h using a relaxation scheme. After few iterations (typically 1 or 2), high-frequency components of the defect are eliminated. Here $\boldsymbol{u}^{(p-1)}$ is the initial value of the iterative scheme.

2. Compute the associated residual $\boldsymbol{r}_h = f_h - A_h\boldsymbol{u}$ and transfer it to a coarser grid Ω^H. We denote by $\boldsymbol{r}_H$ the transferred residual.

3. Relax the so-called defect equation $A_H\boldsymbol{e}_H = \mathbf{r}_H$ on a coarser grid Ω^H. The lower-frequency parts are shifted higher and the smoother can work efficiently again (few iterations are required).

4. Correct the original approximation $\boldsymbol{u}^{(p)} = \boldsymbol{u} + \boldsymbol{e}_h$. Here $\boldsymbol{e}_h$ is obtained by interpolating $\boldsymbol{e}_H$ on the fine grid Ω^h.

A typical choice for H is $2h$. In addition, one may consider the former coarse grid Ω^H as the new fine grid, and apply a coarse-grid correction on the defect equation. Hence, this coarse grid correction can be extended on a grid hierarchy $(\Omega^{2^k h})_{k\in\mathbb{N}}$. Then, the coarse grid correction step is applied recursively in order to smooth even the lowest frequency parts of the defect equation [10]. On the coarsest grid, the defect equation is solved exactly either by using a direct method or by performing few passes of the relaxation scheme (if it has sufficiently good convergence properties on that coarsest grid) [209] . Fortunately, those coarse grid passes are significantly faster than fine grid sweeps since the coarsest grid consists of few points (even one in ideal cases). This leads to recursive algorithms which traverse between fine and coarse grids in the grid hierarchy and corresponds to the common and powerful **V-Cycle** (Figure 2.3). In this figure,

- The coarsest grid level is represented as a square.
- Blue circles denote relaxation sweeps that occur on a specified fine grid level.
- Black arrows restrict the fine grid residual to the coarse grid.

- Conversely, red arrows corresponds to the interpolation of the defect from a coarser grid to a finer one. This interpolated value is added to the corresponding previous fine grid approximation.

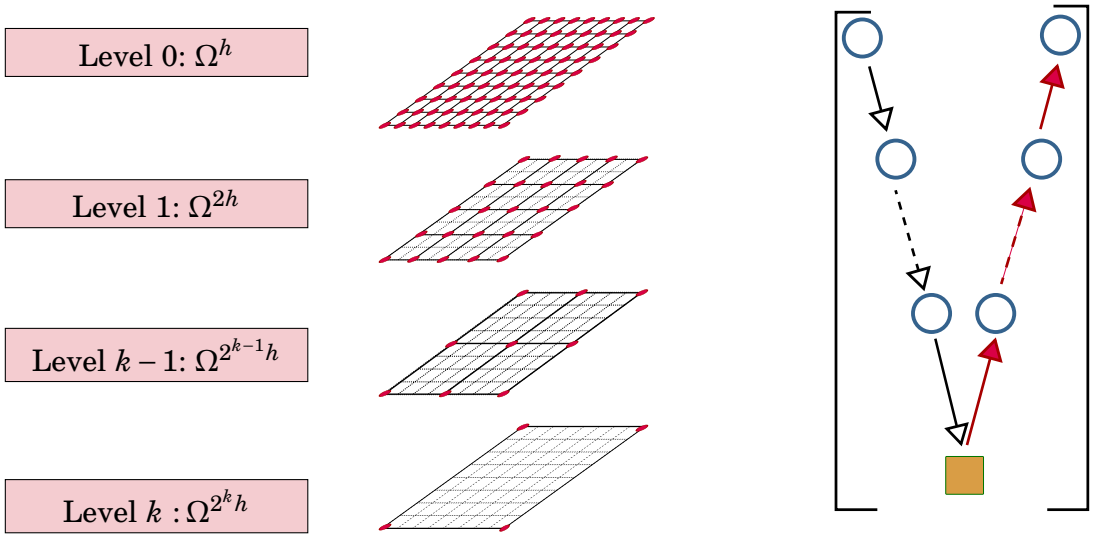


Figure 2.3: Sketch of multigrid V-cycle on a hierarchy of $k+1$ grids. For each grid Ω^{2h}, red dots represent grid points.

The **Full multigrid** algorithm extends the performance of the V-cycle approach by using the coarser grid in order to provide an initial guess to the solution of the finer level. Precisely, one combines a unidirectional multilevel approach with a V-cycle and starts on each level k with an interpolated initial guess computed on the level $k-1$, yielding to the nested iteration. Figure 2.4 depicts a full multigrid algorithm on a hierarchy of 3 grids is used :

- V-Cycles, starting from a specified grid level, are separated by brackets.
- The prolongation of a coarse grid solution from coarse to fine grid is described with blue arrows.

Other variants of multigrid algorithm exist but are not discussed here. For instance, methods using W-cycle are presented in detail in [189, 209]. For this work, Full multigrid with V-cycle strategy was satisfactorily implemented.

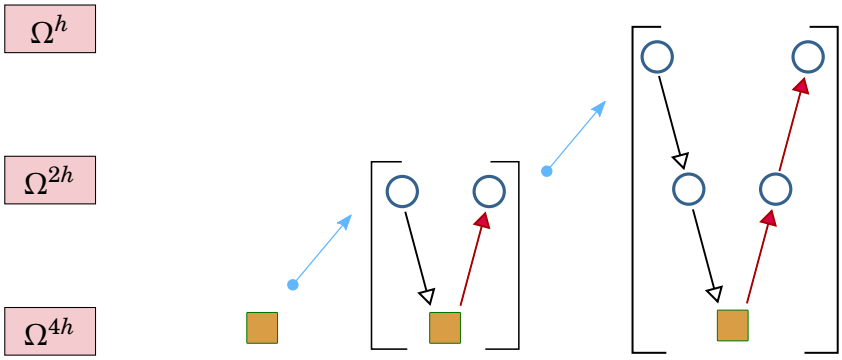
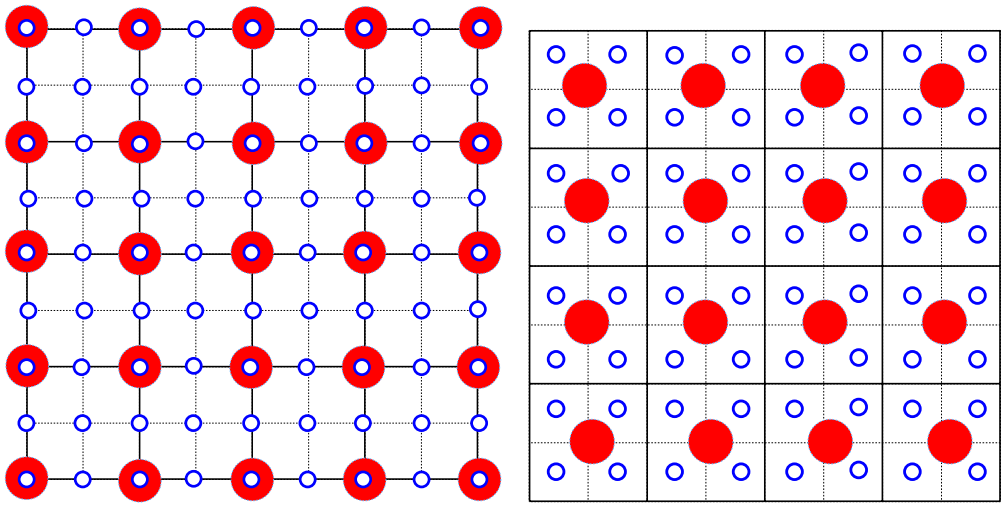


Figure 2.4: Sketch of Full multigrid algorithm with V-cycle:

2.5.3 Coarsening

The discretization of the problem domain Ω presupposes a division into cells which form a resulting grid. However, an important choice to be made in the implementation of the multigrid method is the arrangement of the unknowns within the grid. The unknowns can be placed at the vertices of a grid (**Vertex-Centered coarsening**). Another option is the choice of the unknowns at cell centers (**Cell-Centered coarsening**). The choice of the arrangement of unknowns mainly affects the definition of the operations between different grids level (Figure 2.5). We implemented both vertex-centered and cell-centered strategies. Nevertheless, it is hard to single out which location of unknowns is best in general [209].



(1) Vertex-centered Grid (2) Cell-centered grid

Figure 2.5: Sketch of some discretization strategies employed in multigrid

2.5.4 Inter-grid operations

Multigrid algorithms require going back and forth between several grid problems related to the solution of the same original equations. We define a prolongation operator and a restriction operator in order to switch between different grids.

2.5.4.1 Prolongation

The prolongation $I_{2h}^{h} : \Omega^{2h} \rightarrow \Omega^{h}$ transfers a vector from a domain with grid spacing $2h$ into a domain with a spacing h. This operation can be described through stencil notation (*see* [33, 209]). In case of vertex-centered coarsening, the used stencil is the trilinear interpolation:

$$I_{2h}^{h} = \underbrace{\frac{1}{64}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}}_{bottom} \quad \underbrace{\frac{1}{64}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}}_{center} \quad \underbrace{\frac{1}{64}\begin{bmatrix} 2 & 4 & 2 \\ 4 & 8 & 4 \\ 2 & 4 & 2 \end{bmatrix}}_{top} \tag{2.36}$$

2.5.4.2 Restriction

A restriction operator $R_{2h}^{h} : \Omega^{h} \rightarrow \Omega^{2h}$ maps h-grid functions to $2h$-grid functions. For a vertex-centered coarsening, a trivial restriction operator is the injection operator. In fact, figure 2.5 shows that each node of a coarser grid matches a node from a finer grid. However, we choose to employ another frequently used restriction operator, namely, the full weighting (FW) [209] given by :

$$R_{h}^{2h} = \underbrace{\frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}}_{bottom} \quad \underbrace{\frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}}_{center} \quad \underbrace{\frac{1}{8}\begin{bmatrix} 2 & 4 & 2 \\ 4 & 8 & 4 \\ 2 & 4 & 2 \end{bmatrix}}_{top} \tag{2.37}$$

For a cell-centered coarsening, a typical restriction takes the average of values at neighboring fine grid points; as figure 2.5 shows that every coarse grid point is surrounded by 4 (8) fine grid ones in 2D (3D). Moreover, the expression of inter-grid operations associated to cell-centered coarsening can be found in [117, 148, 209, 223]. In general, the choice of inter-grid operators must be taken with caution. A rule of thumb that arises

from the convergence analysis of multigrid methods, states that these transfer operators must fulfill [33, 209, 224] :

$$m_p + m_r > M, \tag{2.38}$$

where m_p,m_r are order of transfer operators (prolongation and restriction, respectively) and M the order of the partial differential equation. m_p and m_r can be determined by studying the effect of transfer operators on the Fourier components [88]. Also, following [82, 224], there is a straightforward relation between the order of the transfer operator as defined here and the degree of the polynomial that is exactly interpolated by the corresponding interpolation rule. Precisely, an interpolation order is k if the operator leaves all polynomials of degree $k-1$ invariant.

Equation (2.38) is a necessary condition to prove mesh-size independent convergence rates for multigrid following the analysis from [82]. However, in case of cell-centered discretization this condition can be weakened [28].

2.5.5 Defect equation on the coarse grid

The definition of A_H in the defect equation is the last critical point which needs to be discussed. A natural choice is the discretization of the continuous Laplace operator in (2.31) on coarse grids. This approach is called Discretization Coarse Grid Approximation (DCA). An alternative approach is the Galerkin Coarse Grid Approximation (GCA) which uses the so-called *Galerkin product*

$$A_H = R_h^H A_h I_H^h \tag{2.39}$$

Although DCA seems more straightforward, GCA has some advantages as discussed in [209, 213, 224]. Nevertheless, after comparing performances of both approaches on several cases, we cannot afford to give a clear advantage.

Part II

Incremental algorithms

CHAPTER 3

INCREMENTAL EWALD METHODS

Contents

3.1 Overview

Long-range interactions are often taken into account in molecular dynamics simulations. Given a system of N charges q_i located at the position $\boldsymbol{x}_i$ in a cubic box $\mathscr{B}$ of length L, we recall that the electrostatic interaction energy can be written as follows (*see* equation (2.1)):

$$E = \frac{1}{4\pi\epsilon_0} \sum_{\boldsymbol{n}\in\mathbb{Z}^3} \sum_{i=1}^{N} \sum_{j=1}^{N} {}' q_i q_j \frac{1}{||\boldsymbol{x}_i^{\boldsymbol{n}} - \boldsymbol{x}_j||}. \tag{3.1}$$

Here, we assumed that the simulation box $\mathscr{B}$ has periodic boundary conditions (p.b.c). ϵ_0 is the dielectric permittivity and $\boldsymbol{x}_i^{\boldsymbol{n}} = \boldsymbol{x}_i + \boldsymbol{n}L, \boldsymbol{n} \in \mathbb{Z}^3$ corresponds to the replicated particle images. In summation (4.1), we used the $'$ symbol to exclude the divergent terms $j = i$ for $\boldsymbol{n} = \boldsymbol{0}$ [112].

As electrostatics are pairwise long-range interactions, a simple truncation method is not enough to handle them properly. In fact, the computational artifacts introduced by approximate cutoff-based methods aren't suitable for stable MD simulations.

However, popular methods that efficiently compute these interactions, split section 4.1 in short-range and long-range terms [10] in order to overcome the quadratic complexity of a direct particle-particle method. The short-range part should decay fast and be negligible beyond some cutoff distance. Therefore, this component may take advantage of neighbor list algorithms. Meanwhile the treatment of long-range terms depends on the chosen method.

Ewald summation, named after Paul Peter Ewald[1], leads to an $\mathscr{O}(N^{3/2})$ method by surrounding each point charge in the system by a Gaussian distribution of equal magnitude and opposite sign. The counter-charge screens the original potential making it short-ranged and easy to handle. Thus, a second imaginary charge distribution of the same sign as the point charge is added to cancel out the screening charge. Therefore, the smoothness of this distribution can be exploited by summing on the reciprocal space since the corresponding Fourier transform can be evaluated sufficiently accurately in only a few terms.

By using FFT on a gridded version of the Ewald summation, one can significantly reduce the cost of evaluating electrostatics. Although P3M[90], Smoothed Particle Mesh Ewald (SPME)[58], Particle Mesh Ewald (PME)[59] and Fast Fourier Poisson (FFP)

[1]Paul Peter Ewald,(January 23, 1888 in Berlin, Germany – August 22, 1985 in Ithaca, New York) was a German crystallographer and physicist, a pioneer in the study of crystal structure by means of X -ray scattering (http://rsbm.royalsocietypublishing.org/content/roybiogmem/34/133)

methods differ by the number of FFT used, they achieve a $\mathcal{O}(N\log(N))$ computational complexity. These methods are well suited to molecular dynamics simulations where the entire system is perturbed at each step. However, when only a small number of interactions are needed, these methods are limited by the requirement to recalculate FFT on the whole grid. Conversely, Ewald summation might be optimized on ARMD systems where most particles stay at their location at two successive timesteps.

Given two successive timesteps ($t = t_0$) and ($t = t_1$), particles have two kinds of dynamics in ARMD. Whilst the so-called restrained charges are frozen at two successive timesteps, the active ones are allowed to move freely. Our goal is to take advantage of the fact that some charges are restrained from t_0 to t_1 in order to incrementally update the electrostatic potential energy at t_1 knowing its value at t_0, thus speeding up the calculation of long-range forces. After introducing several aspects of the Ewald summation in section 3.2, we will derive its incremental update in section 3.3. Likewise, section 3.5 will present an incremental version of grid-based Fourier methods for electrostatics.

3.2 Ewald Summation in a nutshell

Ewald summation is based on splitting the slowly convergent term $r \mapsto \frac{1}{r}$, in (4.1), into two series which can be computed much faster (for a given accuracy). The trick consists in introducing a splitting function f such that:

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1-f(r)}{r} \tag{3.2}$$

where $r \mapsto \frac{f(r)}{r}$ is a rapidly decreasing function on $\mathbb{R}$. The complementary error function is often chosen as f.

Hence, given a system of N particles in a periodic box, the Ewald summation rewrites the total coulombic interaction energy as follows [6, 52, 54]

$$E = E^{short} + E^{long} - E^{self} \tag{3.3}$$

where

$$E^{short} = \frac{1}{4\pi\epsilon_0}\frac{1}{2}\sum_{\boldsymbol{n}}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{|\boldsymbol{x}_i - \boldsymbol{x}_j + \boldsymbol{n}L|}\operatorname{erfc}\left(\frac{|\boldsymbol{x}_i - \boldsymbol{x}_j + \boldsymbol{n}L|}{\sigma\sqrt{2}}\right) \tag{3.4}$$

$$E^{long} = \frac{1}{2V\epsilon_0}\sum_{\boldsymbol{k}\neq 0}\frac{\exp(-\sigma^2 k^2/2)}{k^2}|S(\boldsymbol{k})|^2 \tag{3.5}$$

$$E^{self} = \frac{1}{4\pi\epsilon_0}\frac{1}{\sqrt{2\pi}\sigma}\sum_{i=1}^{N} q_i^2. \tag{3.6}$$

Here σ denotes the so-called Ewald parameter, V is the volume of the simulation box and $\boldsymbol{k} \in \mathbb{K}^3 = \{2\pi \boldsymbol{n}/L \,|\, \boldsymbol{n} \in \mathbb{Z}^3\}$ is a reciprocal (or $\boldsymbol{k}$-space) vector. $S(\boldsymbol{k})$ corresponds to the structure factor of the charge distribution, namely,

$$S(\boldsymbol{k}) \equiv \sum_{j=1}^{N} q_j e^{i\boldsymbol{k}.\boldsymbol{x}_j}. \tag{3.7}$$

$E^{long} + E^{short}$ includes, for each particle, the self term excluded in the summation (4.1). Therefore, one may remove this unnecessary contribution from the summation (3.3), using the sum E^{self}. Fortunately, E^{self} is a constant, as it does not depend on the atomic positions. Thus, its evaluation can be done once, while starting the MD simulation.

In addition, knowing that E^{self} is constant, the coulombic force acting on a given particle i can be written as

$$\begin{aligned} \boldsymbol{f}_i &\equiv -\frac{\partial E}{\partial \boldsymbol{x}_i} \\ &= -\frac{\partial E^{short}}{\partial \boldsymbol{x}_i} - \frac{\partial E^{long}}{\partial \boldsymbol{x}_i} \end{aligned} \tag{3.8}$$

$$\boldsymbol{f}_i = \boldsymbol{f}_i^{short} + \boldsymbol{f}_i^{long}. \tag{3.9}$$

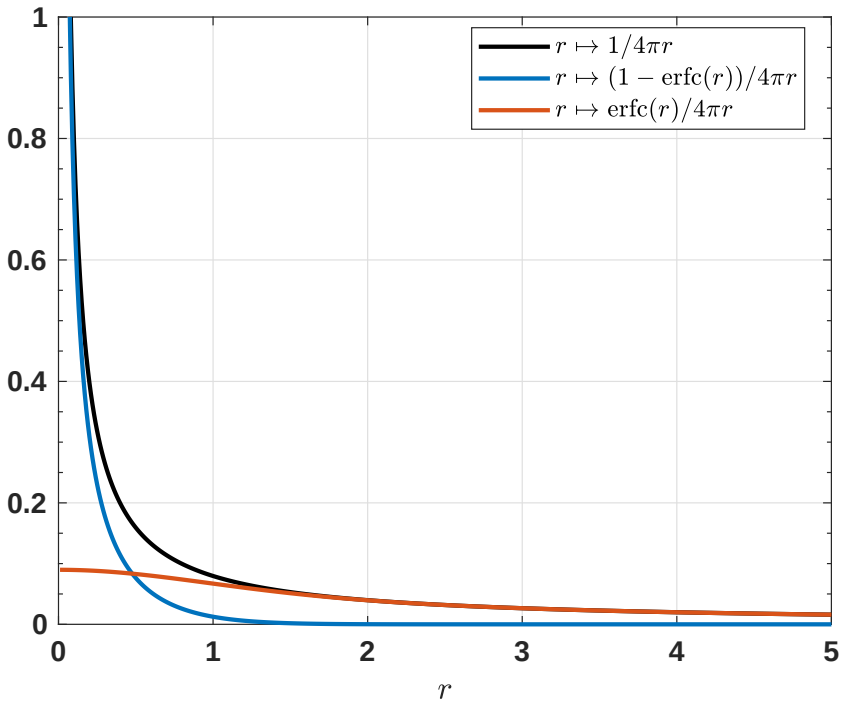


Figure 3.1: Splitting function used in Ewald Summation.

Here, E^{short} accounts for the usual coulombic interaction now multiplied by an error function. This contribution can be computed directly from a sum in real space, since it is essentially due to the pair potential

$$\Phi : r \mapsto \frac{\operatorname{erfc}(r)}{r}. \tag{3.10}$$

Figure 3.1 shows that Φ decays fast and, therefore, is eager to be truncated with a user-specified cutoff distance r_c.

Similarly, it is possible to rapidly evaluate the sum E^{long} in the reciprocal space. This contribution can be truncated with a spherical cutoff, since the quantity $\frac{\exp(-\sigma^2 k^2/2)}{k^2}$ vanishes for large reciprocal space vectors $\boldsymbol{k}$. In other words, E^{long} can accurately be represented by the first few terms of its Fourier series [91]. Therefore, we denote by k_{max} the reciprocal space cutoff.

Furthermore, an estimation of errors induced by these truncations can be found in [112, 163]. Precisely, the expected real-space cutoff error of the forces acting on one particle i follows

$$\delta f_i^{short} = 2|q_i| \left(\frac{Q^2}{r_c L^3} \right)^{1/2} \exp\left[-\frac{r_c^2}{2\sigma^2} \right] \tag{3.11}$$

where the total square sum of charges Q^2 reads

$$Q^2 = \sum_{i=1}^{N} q_i^2. \tag{3.12}$$

Likewise, Kolafa and Perram [112] estimated the standard deviation of the $\boldsymbol{k}$-space cutoff error of the forces acting on charge i as follows

$$\delta f_i^{long} = \frac{|q_i|}{\sigma\sqrt{2}} \frac{\pi^{-3/2}}{V\epsilon_0} \left(\frac{Q^2}{k_{max}} \right)^{1/2} \exp\left[-\frac{\sigma^2 k_{max}^2}{2} \right]. \tag{3.13}$$

By demanding that the real- and reciprocal- space contributions to the error should be equal, they showed that the Ewald parameter and the cutoffs (r_c, k_{max}) can be adjusted such that the prescribed accuracy is achieved with $\mathcal{O}(N^{3/2})$ operations.

3.3 Incremental Ewald Summation

3.3.1 Incremental update of the short-range contribution

The short-range nature of the pair potential Φ (equation (3.10)) suggests the use of neighbor list algorithms. The sum E^{short} can be efficiently determined by introducing, for each particle i, a neighbor list $\mathcal{N}(i)$ which holds neighbors of particle i, namely

$$\mathcal{N}(i) := \left\{ \boldsymbol{x}_p = \boldsymbol{x}_j + \boldsymbol{n}\boldsymbol{L} \;\middle|\; \begin{array}{l} j \in \{1,\dots,N\}; \boldsymbol{n} \in \mathbb{Z}^3 \\ \boldsymbol{x}_p \neq \boldsymbol{x}_i \\ |\boldsymbol{x}_p - \boldsymbol{x}_i| \leq r_c \end{array} \right\}. \tag{3.14}$$

Therefore, E^{short} reads

$$E^{short} = \frac{1}{8\pi\epsilon_0} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}(i)} \frac{q_i q_j}{|\boldsymbol{x}_{ij}|} \operatorname{erfc}\left(\frac{|\boldsymbol{x}_{ij}|}{\sigma\sqrt{2}} \right). \tag{3.15}$$

Here $\boldsymbol{x}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$ is the inter-particle distance.

Moreover, the short-range contribution to electrostatic forces reads [53]

$$\boldsymbol{f}_i^{short} = \frac{1}{4\pi\epsilon_0} \sum_{j \in \mathcal{N}(i)} q_i q_j \frac{\boldsymbol{x}_{ij}}{|\boldsymbol{x}_{ij}|^2} \left[\frac{2}{\sigma\sqrt{2\pi}} \exp\left(\frac{|\boldsymbol{x}_{ij}|^2}{2\sigma^2} \right) - \operatorname{erfc}\left(\frac{|\boldsymbol{x}_{ij}|}{\sigma\sqrt{2}} \right) \right]. \tag{3.16}$$

$\boldsymbol{f}_i^{short}$ can be rewritten as follows

$$\boldsymbol{f}_i^{short} = \sum_{j \in \mathcal{N}(i)} \boldsymbol{f}_{ij}^{short}, \tag{3.17}$$

where $\boldsymbol{f}_{ij}$ denotes the contribution of particle j in the short-range component of the force acting on particle i, namely,

$$\boldsymbol{f}_{ij}^{short} = \boldsymbol{f}_{ji}^{short} = \frac{1}{4\pi\epsilon_0} q_i q_j \frac{\boldsymbol{x}_{ij}}{|\boldsymbol{x}_{ij}|^2} \left[\frac{2}{\sigma\sqrt{2\pi}} \exp\left(\frac{|\boldsymbol{x}_{ij}|^2}{2\sigma^2} \right) - \operatorname{erfc}\left(\frac{|\boldsymbol{x}_{ij}|}{\sigma\sqrt{2}} \right) \right]. \tag{3.18}$$

$\boldsymbol{f}_i^{short}$ can be seen as a force induced by a short-range pairwise interaction. Consequently, during ARMD simulations, an incremental update of $\boldsymbol{f}_i^{short}$ and E^{short} may follow the pattern introduced in [191]. Particles in ARMD are, at each timestep, either **active** or **restrained** at their location. For the moment, let us assume that particles do not change their state during the simulation.

Thus, the main ideas of an incremental update of the short-range component of electrostatic forces follow:

- If i is an active particle, as its position changes over time, the value $\boldsymbol{f}_{ij}^{short}$ varies at each timestep, regardless of the state of the particle j. The total short-range force needs to be recomputed.
- However, if i is a restrained particle, the value $\boldsymbol{f}_{ij}^{short}$ remains unchanged between two successive timesteps, for a particle j which is restrained. In this case, one can save a substantial amount of CPU time by avoiding the re-computation of $\boldsymbol{f}_{ij}^{short}$. Conversely, this value might change when j is active.

In the light of the above, for a restrained particle i, the short-range force $\boldsymbol{f}_i^{short}$ in (3.17) can be split in two sums: one over active neighbors of i $\left(\boldsymbol{f}_i^A\right)$; the other over its restrained neighbors $\left(\boldsymbol{f}_i^R\right)$:

$$\boldsymbol{f}_i^{short} = \boldsymbol{f}_i^A + \boldsymbol{f}_i^R \tag{3.19}$$

$$= \sum_{\mathcal{N}(i)\cap active} \boldsymbol{f}_{ij}^{short} + \sum_{\mathcal{N}(i)\cap restrained} \boldsymbol{f}_{ij}^{short}. \tag{3.20}$$

Here, «*active*» and «*restrained*» denote, respectively, the list of active or restrained particles. $\boldsymbol{f}_i^R$ can be cached and reused. Only $\boldsymbol{f}_i^A$ needs to be computed.

Algorithmic speedup — Assuming that particles are uniformly distributed in the simulation box, one can easily estimate the algorithmic speed up of the presented strategy. In fact, at each timestep, the computational cost of the incremental update of short-range forces depends on $\alpha_a \in [0,1]$, the ratio of active particles. An evaluation of $\boldsymbol{f}_i^{short}$ using the summation (3.17) has a constant cost per particle. This cost is related to the size of the neighbor list. The total cost of this summation is, in terms of unitary evaluations of $\boldsymbol{f}_{ij}^{short}$,

$$C_0 = Nm \tag{3.21}$$

where N is the number of particles and m the average size of neighbor lists $\mathcal{N}(i)$. Here, we deliberately do not use Newton's third law which would reduce C_0 by a factor 2, since it wouldn't alter the conclusion of this analysis.

The evaluation's cost of $\boldsymbol{f}_i^{short}$ for all $N_a = \alpha_a N$ active particles is $C_a = \alpha_a Nm$. Meanwhile, the number of its active neighbors is $\alpha_a m$ for a given restrained particle. Therefore, the computation of $\boldsymbol{f}_i^A$ will require $C_r = \alpha(N - N_a)m$ evaluations of $\boldsymbol{f}_{ij}^{short}$. The total cost of the incremental update of $\boldsymbol{f}_i$ follows

$$C_1 = Nm(2 - \alpha_a)\alpha_a. \tag{3.22}$$

The related speedup reads

$$Sp = \frac{C_0}{C_1} \tag{3.23}$$

$$= \frac{1}{(2-\alpha_a)\alpha_a}$$

$$= \frac{1}{1-\alpha_r^2} \tag{3.24}$$

where $\alpha_r = 1 - \alpha_a$ is the fraction of restrained particles.

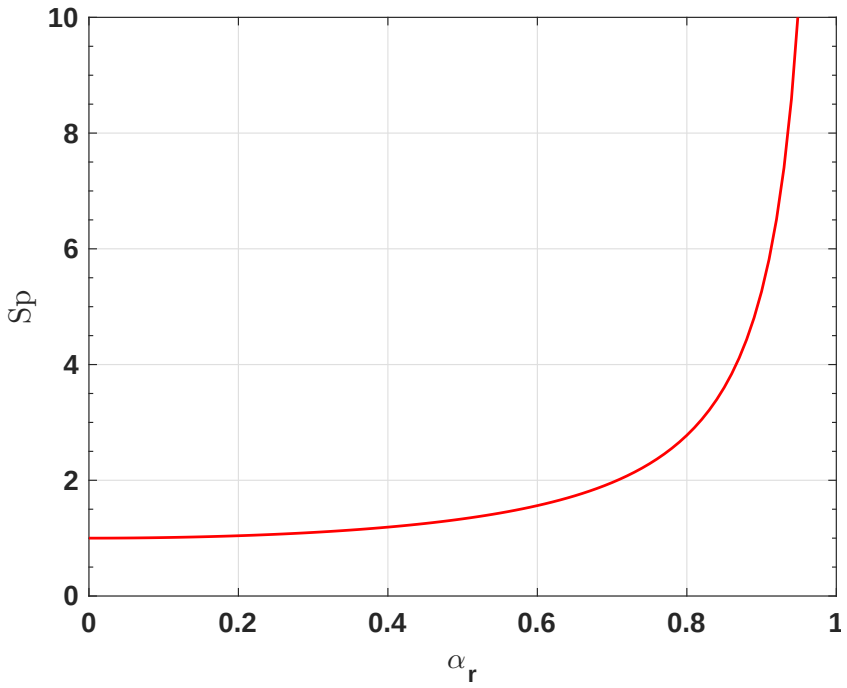


Figure 3.2: Algorithmic speedup of the incremental update of short-range contributions in Ewald summation: The fraction of restrained particles is $\alpha_r = 1 - \alpha_a$. In this ideal case, particles cannot switch their state.

When more than 70% of particles are restrained the proposed method reduced by a factor 2 the cost of updating $\boldsymbol{f}_i^{short}$. Moreover, a 10× speedup is achieved for fewer than 5% active particles (*see* figure 3.2).

However, while performing ARMD simulations, it is likely that a few particles switch state (to active or restrained) at each timestep . In order to correctly retrieve short-range contributions, a slight modification is needed before updating particle positions. Basically, when a particle p switches from restrained to active, its restrained component $\boldsymbol{f}_p^R$ has to be cleared, since the total short-range force has to be fully recomputed. In addition, all its contributions to its neighbors forces must be subtracted. Since the short-range

forces of its active neighbors have to be recomputed anyway, one may loop only over the restrained neighbors $j \in \mathcal{N}(p) \cap restrained$ of the switched particle p and remove from each $\boldsymbol{f}_j^R$ the term $\boldsymbol{f}_{pj}^{short}$. Similarly, when a particle p switches from active to restrained, one may store its restrained component $\boldsymbol{f}_p^R$ by looping over its restrained neighbors. While doing so, the term $\boldsymbol{f}_{pj}^{short}$ has to be added to $\boldsymbol{f}_j^R$. The extra cost induced by this modification depends linearly on the amount of switched particles. Therefore, it has a minor impact on the overall speedup.

A more general analysis of the time complexity of the incremental update of short-range interactions can be found in [191]. The obtained results are close to the ideal case presented in figure 3.2. This analysis takes into account the construction and update of specialized neighbor lists that hold for each particle, only its active neighbors ($\mathcal{N}(p) \cap active$). These lists are called *active neighbor lists* and have been successfully evaluated for the incremental update of short-range potentials [190]. In addition, they can be used in order to enable Newton's third law [190].

Finally, it is straightforward that the incremental update of the short-range energy follows the principles presented above. Thus, its derivation can be obtained with ease.

3.3.2 Incremental update of the long-range contribution

The evaluation of the long-range component E^{long} in (3.3) requires the computation of the structure factor $S(\boldsymbol{k})$. The incremental update of the structure factor can be derived using a divide-and-conquer paradigm, namely,

$$\begin{aligned} S(\boldsymbol{k}) &= S^{(a)}(\boldsymbol{k}) + S^{(r)}(\boldsymbol{k}) \\ &= \sum_{active} q_j e^{i\boldsymbol{k}.\boldsymbol{x}_j} + \sum_{restrained} q_j e^{i\boldsymbol{k}.\boldsymbol{x}_j}. \end{aligned} \tag{3.25}$$

Then, we take advantage of the fact that a given accuracy can be achieved with only a few terms $S(\boldsymbol{k})$. The restrained structure factor $S^{(r)}(\boldsymbol{k})$ can be held in memory and updated with respect to the few particles that switch their state. Therefore, the cost of the incremental update of the structure factor requires $\mathcal{O}(N_a + N_s)$ operations, where N_s is the number of switched particles.

Usually, the term $\frac{\exp(-\sigma^2 k^2/2)}{k^2}$ is saved at the beginning of the simulation. Therefore, the summation (3.3) cannot be optimized further.

The incremental update of the long-range component of the energy can be extended to electrostatics forces. The long-range contribution $\boldsymbol{f}_i^{long}$ can be computed by evaluating

the derivative of the structure factor

$$\frac{\partial S(\boldsymbol{k})}{\partial \boldsymbol{x}_i} = i\boldsymbol{k} q_p e^{i\boldsymbol{k}.\boldsymbol{x}_i}. \tag{3.26}$$

From (3.9)

$$\begin{aligned} \boldsymbol{f}_i^{long} &= \frac{1}{2V\epsilon_0} \sum_{\boldsymbol{k}\neq 0} \frac{\exp(-\sigma^2 k^2/2)}{k^2} \left(\overline{S(\boldsymbol{k})} \frac{\partial S(\boldsymbol{k})}{\partial \boldsymbol{x}_i} + S(\boldsymbol{k}) \frac{\partial \overline{S(\boldsymbol{k})}}{\partial \boldsymbol{x}_i} \right) \\ &= \frac{q_i}{V\epsilon_0} \sum_{\boldsymbol{k}\neq 0} \frac{e^{-\sigma^2 k^2/2}}{k^2} \boldsymbol{k} (\sin(\boldsymbol{k}.\boldsymbol{x}_i)\mathbb{R}\mathrm{e}(S(\boldsymbol{k})) - \cos(\boldsymbol{k}.\boldsymbol{x}_i)\mathbb{I}\mathrm{m}(S(\boldsymbol{k}))). \end{aligned} \tag{3.27}$$

In this summation, the evaluation of the structure factor, can be sped up following the splitting in (3.25). Furthermore, one may accelerate the evaluation of $\boldsymbol{f}_i^{long}$ for restrained particles by storing both $\sin(\boldsymbol{k}.\boldsymbol{x}_i)$ and $\cos(\boldsymbol{k}.\boldsymbol{x}_i)$. By this means, the unnecessary pricey evaluation of trigonometric functions is avoided.

The main drawback of this strategy comes from the substantial amount of memory it requires for holding these terms. Unlike the energy, the forces require for each atom a separate set of data . The memory requirement of this strategy is thus proportional to the product of the number of particles and the number of needed reciprocal vectors.

3.3.3 Evaluation of the method

We evaluate the incremental Ewald summation method on a SPC/E water system of 36000 particles. A primary configuration of the system was generated by replicating (twice per dimension) an original 4500 particles system downloaded from [118]. Then, a quick minimization was performed in order to remove eventual clashes. The minimized structure defines the initial configuration.

First, various parameterizations of the Ewald splitting were tested. Basically, the target accuracy was set up to 10^{-4} relative force error. Then, a cutoff radius is defined between 10 Å and 18 Å. The accuracy setting is used in conjunction with the real space cutoff to determine the number of $\boldsymbol{k}$-space vectors. Used configurations are listed in table 3.1. In the same table, the CPU times required by the main components of the traditional Ewald summation at each timestep are shown. T_{real} denotes the time required for the evaluation of short-range forces as well as the short and self contributions to the electrostatic energy. Similarly, $T_{\boldsymbol{k}-space}$ refers to CPU time consumption of the reciprocal space calculations.

Increasing the real-space cutoff r_{cut}, reduces $T_{\boldsymbol{k}-space}$ while increasing T_{real}. The best configuration was found with a 16 Å cutoff and 9^3 $\boldsymbol{k}$-space vectors. This configuration

leads to a total time per timestep $T_{total} = 1.34\,\text{s}$. The ratio between real- and reciprocal-space timings is equal to 1.85.

r_{cut} [Å]	σ [Å^{-1}]	k_{max}	T_{real} [s]	$T_{\boldsymbol{k}-space}$ [s]	T_{total} [s]
10	0.26	15	0.21	2.16	2.37
11	0.23	14	0.31	1.73	2.04
12	0.21	13	0.39	1.38	1.77
13	0.20	12	0.48	1.08	1.56
14	0.18	11	0.60	0.82	1.42
15	0.17	10	0.72	0.63	1.35
16	0.16	9	0.87	0.47	1.34
17	0.15	9	1.00	0.47	1.47
18	0.14	8	1.09	0.39	1.48

Table 3.1: Performance of Ewald summation on a system of SPC/E water

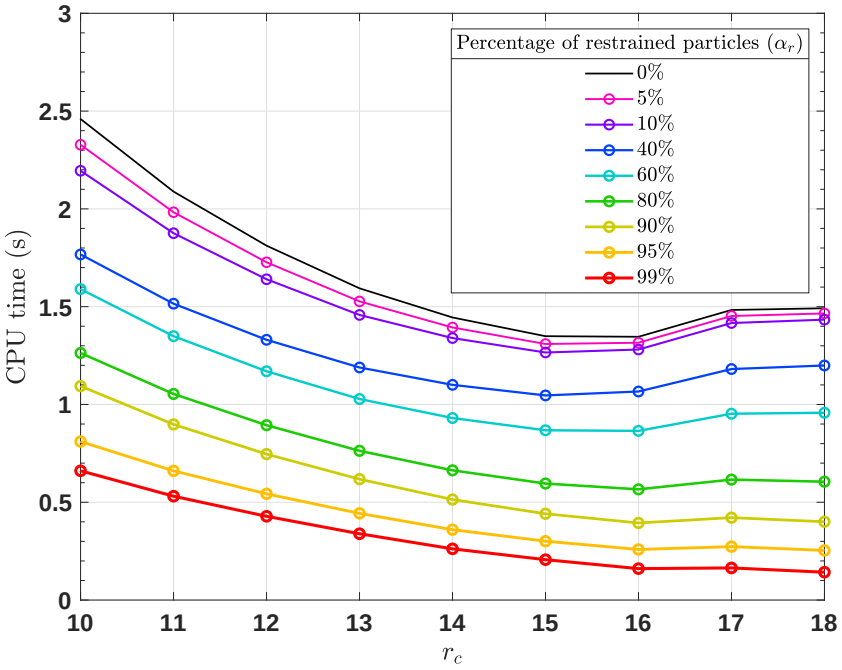


Figure 3.3: Performance of Incremental Ewald Summation.

Then, the water system was partially restrained by freezing some particles for few timesteps. These particles were randomly selected in the simulation box and the percentage of frozen or restrained particles α_r ranges from 0 to 99%. The selected particles are kept frozen for 10 timesteps (no switched particle), in order to average timings. For a given α_r, several independent runs were conducted in order to remove the selection bias. The calculations of electrostatics quantities are performed with the incremental Ewald summation. The averaged total timing is shown in figure 3.3, where the CPU time of the method is plotted against the chosen radius cutoff. While increasing the amount of frozen particles, the presented method reduces well the CPU cost of electrostatics (Figure 3.3). For all tested configurations, the proposed method is more efficient than Ewald technique. For 90% of restrained particles, the incremental Ewald summation is more than 3 times faster than the classical Ewald summation (Figure 3.4).

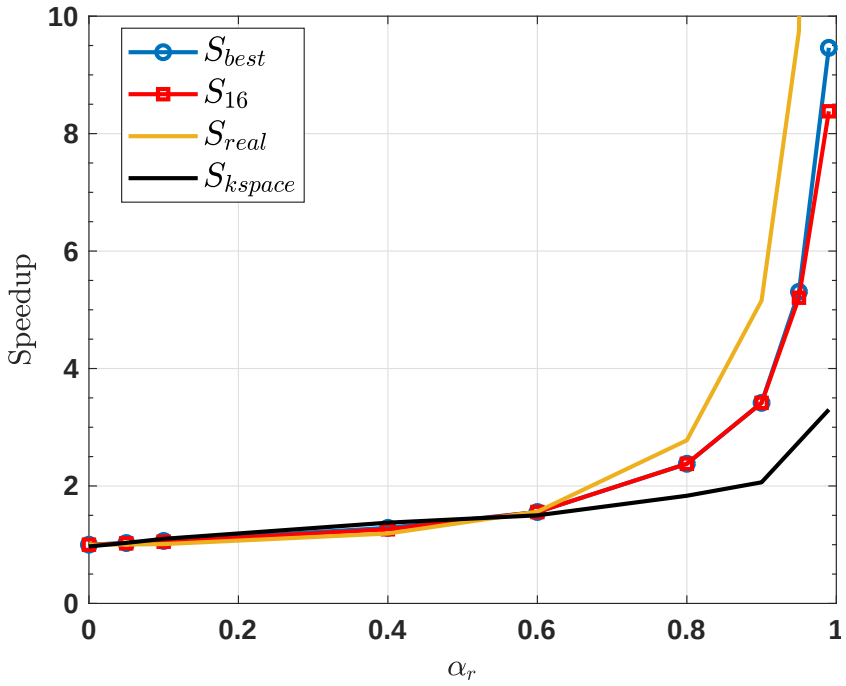


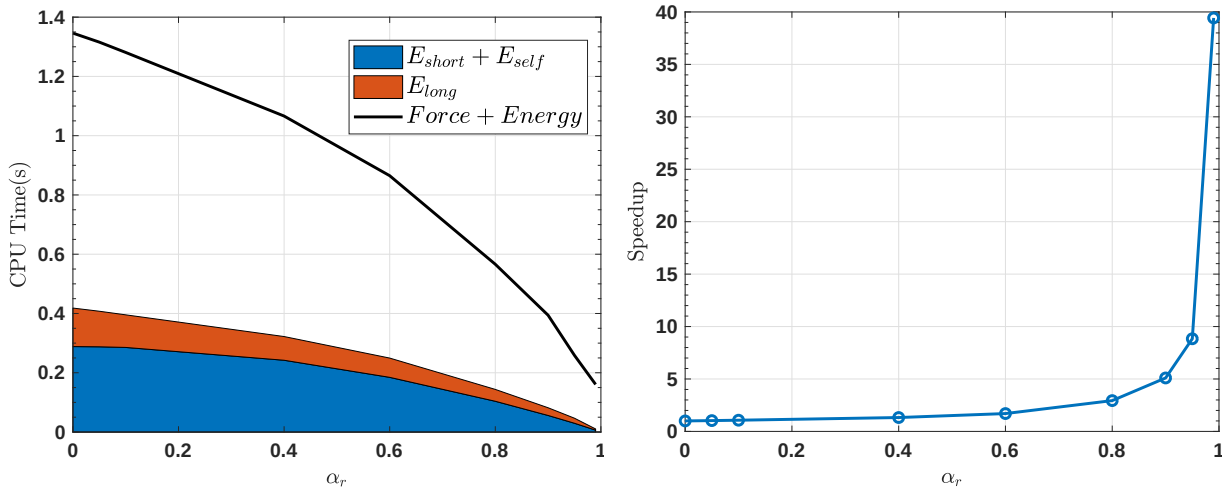
Figure 3.4: Speedup of Incremental Ewald Summation for various ratio of restrained particles

In figure 3.4, S_{best} corresponds to the ratio between the best timing, achieved while varying the cutoff r_{cut} and using incremental Ewald summation, and the best timing achieved with the classical Ewald method.

$$S_{best} = \frac{T_{best}^{incremental}}{T_{best}^{Ewald}} \tag{3.28}$$

where, for a given percentage of restrained particles, T_{best}^{Ewald} is the best total timing achieved with classical Ewald summation and $T_{best}^{incremental}$ is the best total timing achieved using incremental Ewald summation. S_{16} corresponds to the speedup achieved while using a 16 Å cutoff. In addition, speedups of real and reciprocal space calculations are plotted.

A thorough analysis of the performance of the method shows that the fastest configuration is not unique for all percentage of restrained particles. Although $r_{cut} = 16$ was often the best choice, one may prioritize large real-space cutoffs for low percentages of active particles. This comes from the fact that, for a high amount of restrained particles, the incremental evaluation of real space terms ($\boldsymbol{E}^{short}$, $\boldsymbol{f}^{short}$) is more successful than the reciprocal space incremental calculations. Precisely, for less than 1% of active particles, one may achieve a ×3 speedup in reciprocal space S_{kspace} while exceeding ×10 speedup in the real space S_{real}. The performance of the reciprocal space calculations is limited by the fact that the force component $\boldsymbol{f}_p^{long}$ requires for each particle, an unavoidable summation over the k_{max}^3 reciprocal space vectors (3.27).



(1) Timing of the incremental evaluation of energy. The real space cost is shown in blue while the reciprocal cost is in red. The timing including incremental forces is shown as reference.

(2) Speedup of the incremental evaluation of energy (only).

Figure 3.5: Performance of incremental Ewald summation.

Nevertheless, if we restrict ourselves to energy calculations, the presented method shows tremendous performances. Figure 3.5 shows that the cost of energy calculations is reduced by a factor 5 when 90% of particles are restrained. When less than 1% of particles are active one may fairly expect a ×39 speedup. This behavior could be worthwhile in

Markov chain Monte Carlo simulations, where the acceptance or the rejection of a trial move of a randomly chosen particle (or molecule) depends on the difference in energy between the initial configuration and the new one.

3.4 Ewald summation on a mesh

Particle mesh methods are closely related to the Ewald summation method. These methods use meshes in order to accelerate the computation of the reciprocal space contribution in the Ewald splitting. On the other hand, Ewald summation and Particle mesh methods share the same scheme for the real space contribution. Before deriving an incremental treatment of Ewald summation on a mesh, we can gain some more insight on the evaluation of the reciprocal term by rewriting

$$
\begin{aligned}
E^{long} &= \frac{1}{2V\epsilon_0}\sum_{\boldsymbol{k}\neq 0}\frac{\exp(-\sigma^2k^2/2)}{k^2}|S(\boldsymbol{k})|^2 \\
&= \frac{1}{2V\epsilon_0}\sum_{\boldsymbol{k}\neq 0}\frac{\exp(-\sigma^2k^2/2)}{k^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{N} q_i q_j e^{i\boldsymbol{k}.(\boldsymbol{x}_i-\boldsymbol{x}_j)}\right)
\end{aligned} \tag{3.29}
$$

since $|S(\boldsymbol{k})|^2 = S(\boldsymbol{k})\overline{S(\boldsymbol{k})} = S(\boldsymbol{k})S(-\boldsymbol{k})$. Consequently,

$$
E^{long} = \frac{1}{2}\sum_{i=1}^{N} q_i \Phi^{long}(\boldsymbol{x}_i) \tag{3.30}
$$

where $\Phi^{long}(\boldsymbol{x})$ defines the long-range interaction potential, namely,

$$
\Phi^{long}(\boldsymbol{x}) = \frac{1}{V\epsilon_0}\sum_{\boldsymbol{k}\neq 0}\sum_{j=1}^{N}\frac{q_j}{k^2}e^{i\boldsymbol{k}.(\boldsymbol{x}-\boldsymbol{x}_j)}e^{-\sigma^2k^2/2}. \tag{3.31}
$$

Let $\tilde{g}(\boldsymbol{k}) := 1/k^2$ the Fourier transformed Green's function of the Coulomb potential $1/r$. Similarly, $\tilde{\zeta}(\boldsymbol{k}) := e^{-\sigma^2k^2/2}$. Then, we can rewrite $\Phi^{long}(\boldsymbol{x})$ as follows

$$
\Phi^{long}(\boldsymbol{x}) = \frac{1}{V}\sum_{\boldsymbol{k}\neq 0}\left(\tilde{g}(\boldsymbol{k})\tilde{\zeta}(\boldsymbol{k})\tilde{\rho}(\boldsymbol{k})\right)e^{i\boldsymbol{k}.\boldsymbol{x}} \tag{3.32}
$$

where

$$
\tilde{\rho}(\boldsymbol{k}) = \sum_{j=1}^{N} q_j e^{-i\boldsymbol{k}.\boldsymbol{x}_j}. \tag{3.33}
$$

corresponds to the Fourier transform of the distribution of point charges located at atomic positions (*see* equation (2.24)). Interestingly $\tilde{\zeta}(\boldsymbol{k})\tilde{\rho}(\boldsymbol{k})$ is the Fourier transform of

a summation of Gaussian densities located at atomic positions (*see* equation (2.29)). This is a consequence of the fact that Ewald summation smears the Dirac densities created by the point charges by Gaussians.

Furthermore, the long-range interaction potential in (3.32) can be seen as inverse Fourier transform of the following product in the reciprocal space:

$$\tilde{\Phi}^{long}(\boldsymbol{k}) = \tilde{g}(\boldsymbol{k})\tilde{\zeta}(\boldsymbol{k})\tilde{\rho}(\boldsymbol{k}). \tag{3.34}$$

The long-range term of the potential from a given charge distribution can thus be determined with two Fourier transforms. One should notice that this formulation corresponds to a classical resolution of a Poisson equation by Fourier methods (*see* chapter 2). Particle Mesh methods reduce the cost of these Fourier transforms by using FFTs which would require a gridded version of equations (3.33) and (3.34).

Let us define Ω_h as a mesh of spacing h. The corresponding number of mesh points in each direction is $M = \dfrac{L}{h}$. It is possible to replace the continuous charge distribution ρ by a mesh-based charge density ρ_g. At each grid point $\boldsymbol{x}_p$, the mesh-based charge density is given by :

$$\rho_g(\boldsymbol{x}_p) = \frac{1}{h^3}\sum_{i=1}^{N} q_i W(\boldsymbol{x}_p - \boldsymbol{x}_i). \tag{3.35}$$

Here, the summation runs over all particles (in the simulation box) with respect to the periodic boundary conditions. In addition, following [54], the so-called charge assignment W function is classified according to its order P that corresponds to the number of grid points – per dimension – on which each charge is distributed. W is a function of weight one. Usually, one chooses W as a cardinal B-spline with a compact support $\left[-\dfrac{Ph}{2}, \dfrac{Ph}{2}\right]^3$ [17]. In addition, P is an integer between 2 and 8. For instance, the default value of P is 5 in LAMMPS.

Furthermore, we may retrieve the corresponding long-range potential on the grid as follows

$$\Phi_g^{long}(\boldsymbol{x}_p) = \overleftarrow{\mathrm{FFT}}\left[\ \overrightarrow{\mathrm{FFT}}[\rho_g] \times \tilde{G}_{opt}\right](\boldsymbol{x}_p) \tag{3.36}$$

Here, we distinguish the forward $\overrightarrow{\mathrm{FFT}}$ and the inverse $\overleftarrow{\mathrm{FFT}}$. $\tilde{G}_{opt}$ plays the role of the mesh-based version of the screened Fourier transformed Green's function $\tilde{g}(\boldsymbol{k})\tilde{\zeta}(\boldsymbol{k})$. However, as suggested by Holm et al. [53], the determination of $\tilde{G}_{opt}$ is far from trivial. Using directly the *continuum* Green's function does not provide the best approximation of

Φ^{long}. Therefore, the Green's function must be modified in order to ensure that the mesh-based potential coincides, as closely as possible, with the original continuum solution. Moreover, the expression of G_{opt} is influenced by the choice of the assignment function W, leading to a wide range of options [54, 59, 90, 90]. In practice, the optimal Green's function $\tilde{G}_{opt}$ is computed once at the beginning of the simulation.

From the mesh-based long-range potential, one may interpolate the value of the potential at the particles coordinates. The traditional interpolation formula uses the charge assignment function:

$$\Phi^{long}(\boldsymbol{x}_i) \simeq \sum_{\boldsymbol{x}_p \in \Omega_h} \Phi_g^{long}(\boldsymbol{x}_p) W(\boldsymbol{x}_p - \boldsymbol{x}_i) \tag{3.37}$$

Although this summation extends over the complete mesh, it can be evaluated efficiently since a P^{th} order W is non-nil on, at most, P^3 grid points.

Moreover, forces are determined by evaluating the electric field $\mathscr{E}$, which is the gradient of the long-range potential

$$\boldsymbol{f}_i^{long} = q_i \mathscr{E}(x_i). \tag{3.38}$$

First, one may determine the electric field $\mathscr{E}$ with an analytical differentiation [58]. In fact, equation (3.37) can be differentiated analytically by determining the gradient of the assignment function

$$\mathscr{E}(\boldsymbol{x}_i) = \sum_{\boldsymbol{x}_p \in \Omega_h} \Phi_g^{long}(\boldsymbol{x}_p) \nabla W(\boldsymbol{x}_p - \boldsymbol{x}_i). \tag{3.39}$$

This method which determines forces with one backward and one forward FFTs, is called the analytic differentiation. The main drawback of this approach (here called AD-P3M)is the fact that it introduces aliasing errors at atomic positions which leads to the violation of the Newton's third law. The errors in forces can be compensated by removing a term corresponding to self forces [17, 195].

On the other hand, the electric field on the mesh can be evaluated by a differentiation in the Fourier space which reads

$$\mathscr{E}(\boldsymbol{x}_p) = \overleftarrow{\mathrm{FFT}} \left[i\boldsymbol{k} \overrightarrow{\mathrm{FFT}}[\rho_g] \times \tilde{G}_{opt} \right] (\boldsymbol{x}_p) \tag{3.40}$$

Thus, forces are computed with the following interpolation formula

$$\boldsymbol{f}_i^{long} \simeq q_i \sum_{\boldsymbol{x}_p \in \Omega_h} \mathscr{E}(\boldsymbol{x}_p) W(\boldsymbol{x}_p - \boldsymbol{x}_i). \tag{3.41}$$

Hockney and Eastwood [90] demonstrate that this approach (so-called $i\boldsymbol{k}$-P3M) obeys Newton's third law. However, $i\boldsymbol{k}$-differentiation requires three forward and one backward FFTs in order to retrieve forces. Oddly, $i\boldsymbol{k}$-P3M is often more efficient than AD-P3M because the latter usually requires a finer grid in order to obtain the same precision. It is important to realize that the choice of the differentiation method affects the optimal influence function $\tilde{G}_{opt}$ If interested, the expression of the optimal influence function corresponding to $i\boldsymbol{k}$-P3M is available in appendix A.2.

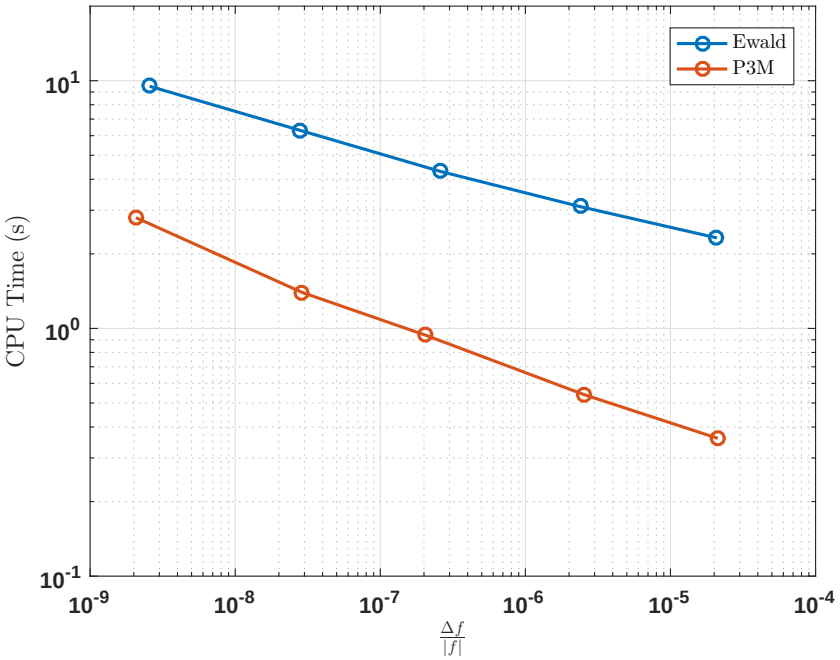


Figure 3.6: Ewald summation vs P3M on a SPC/E water system (36000 charges)

Remark 3.1. *One may evaluate E^{long} without any reverse Fourier transform. In fact, the structure factor $S(\boldsymbol{k})$ is equal to $\tilde{\rho}(\boldsymbol{k})$. Therefore, an approximation of E^{long} is derived by replacing $S(\boldsymbol{k})$ by the discrete Fourier transform of $\tilde{\rho}(\boldsymbol{k})$. Once again, the approximation error is minimized by replacing the term $\exp(-\sigma^2 k^2/2)/k^2$ by $\tilde{G}_{opt}(\boldsymbol{k})$ in summation (3.29). This approach is prioritized by most MD software since it is cheaper than equation (3.30) which requires an explicit evaluation of the interaction potential Φ^{long} through FFT.*

As shown in figure 3.6, P3M is a significant enhancement to Ewald summation, especially on large systems. Therefore, most MD softwares use P3M and its variants. The upcoming derivation of incremental Particle Mesh methods is based on P3M. Nonetheless, one can derive with ease an extension to other Particle Mesh techniques such as SPME and PME.

Before deriving incremental P3M, we may summarize the various steps of the P3M algorithm as follows

1. Compute the short ranged terms, as in Ewald summation.
2. Evaluate the mesh-based charge density $\rho_g(\boldsymbol{x}_p)$.
3. Evaluate its forward FFT of ρ_g.
4. Using the modified Coulomb Green's function $\tilde{G}_{opt}$, determine the Fourier transform of the potential field on the mesh.
5. Finally, interpolate forces at particles' position by applying backward FFTs.

3.5 Incremental Particle Mesh methods

As the incremental evaluation of the short-ranged terms (step 1) has been covered in section 3.3.1, here, the aim is to reduce the cost of the evaluation of long-range quantities $\tilde{\Phi}^{long}$, $\boldsymbol{f}_i^{long}$ when only a few particles update their positions between two consecutive timesteps.

Let us denote by (P_0) our initial problem which determines at each timestep the long-range forces $\boldsymbol{f}_i^{long}$ acting on a system involving restrained particles that do not update their coordinates. At a given timestep $t = k$, we denote by R_k the set of restrained particles while A_k defines the collection of active particles.

Unsuccessful attempts – Basically, the step 2 of P3M can be evaluated incrementally by using the divide-and-conquer strategy employed for the structure factor (*see* equation (3.25) of section 3.3.2). While splitting the mesh-based charge density into *active* and *restrained* contributions, one can compute separately the corresponding potentials. Since all the operators employed by P3M are linear, one can determine the needed potential (or forces) by summing the «*restrained*» potential and the «*active*» potential.

Furthermore, when no particle switches, the restrained potential can be computed once and cached at each following timestep. Unfortunately, the stated strategy is limited by the fact that the most expensive steps (3, 4), which corresponds to FFTs evaluations, weren't accelerated at all.

When the number of active particles is very small, the «*active*» charge density is nil on many grid points. Thereby, it could be attractive, for the forward FFT in step 3, to employ specialized FFT algorithms that might take advantage of this sparsity [74, 87].

Nevertheless, it is neither obvious nor obviously true that the three remaining backward FFTs – needed for forces – could benefit on these family of algorithms. In particular, the output of the first forward FFT populates (almost) all frequencies. Another drawback of these specialized FFTs (which are nowadays popular for Big Data) is the fact that they require a very sparse data. Unfortunately, since each particle affects at least P^3 grid points, the critical point where traditional FFT algorithms are better, is reached quickly and is impractical even for most ARMD simulations. Therefore, we need to define a more efficient strategy that could accelerate P3M.

Successful attempts – When no particle switches, it seems natural to separate the influence of restrained particles from the influence of active ones. In particular, one may calculate and store the restrained contribution to long-range quantities. Thus, the remaining challenge would be to determine the influence of the active particles on the whole system. Two separate meshes can be used and intuitively, the corresponding grid resolutions might depend on the number of particles involved. Precisely, a finer grid (Ω_{h_1}) could be used for the quantities induced by restrained particles and while a coarser grid (Ω_{h_2}) would be dedicated to active ones. In addition, the computations on these grids can be performed asynchronously. The coarser grid has to be updated at each timestep whereas the finer grid computation may occur once (or less frequently, in case of switched particles). If the coarser grid is coarse enough, the corresponding computations will be cheaper than a traditional P3M. The reader should realize that the evaluation of FFTs scales as $\mathcal{O}\left(M^3 \log(M)\right)$ on a M^3 mesh.

With this in mind, our initial problem (P_0) is decomposed in two sub-problems.

- The first problem (P_1) would be to determine the electric field (or the potential or forces) associated to R_0, the collection of restrained particles at the initial timestep $t = 0$. The gathered contribution can be saved in memory. In other words, (P_1) determines the electric field corresponding to $\rho[\mathsf{R}_0]$ the charge density associated to particles in R_0. Particles which are not in R_0, act like virtual particles with no incidence on the electric field. However, forces, here denoted by $\boldsymbol{f}_i^{(1)}$, must be evaluated at the location of all particles by interpolating the computed electric field. Fortunately, we can define an interpolation formula based on the assignment function associated to the mesh used for (P_1). (*see* algorithm 1)

- At each timestep, the problem (P_2) solves the electric field induced by the charge density $\rho[\mathsf{A}_0]$ generated by the active particles (A_0). Here, restrained particles do

not affect the electric field. However, once again, the electric field must be evaluated at all atomic positions in order to retrieve forces $\boldsymbol{f}_i^{(2)}$.

At each timestep, the calculated quantities can be summed up with the cached contribution of active particles (*see* algorithm 2).

$$\rho = \rho_1 + \rho_2 \implies \boldsymbol{f}_i^{long} = \boldsymbol{f}_i^{(1)} + \boldsymbol{f}_i^{(2)} \tag{3.42}$$

where $\rho_1 = \rho[\mathsf{R}_0]$ and $\rho_2 = \rho[\mathsf{A}_0]$.

However, this paradigm doesn't take into account the fact some particles may switch state. This general case would be discussed later in section 3.5.2. The underlying idea will use a modification of the sub-problem (P_2) in order to guarantee the equality (3.42). Nevertheless, for the moment, let restrict ourselves to the following assumption:

«**Throughout the simulation, the distribution of restrained/active particles is fixed**».

$$\forall k \in \mathbb{N},\ \mathsf{A}_k = \mathsf{A}_0 \text{ and } \mathsf{R}_k = \mathsf{R}_0. \tag{A_1}$$

The key ideas underlying the proposed strategy are summarized by figures 3.7 and 3.8.

Algorithm 1 P3M for problem (P_1)

1: **for each** particle i **do**
2: Determine assignment coefficients on the mesh Ω_{h_1} $\left(W_{ip}^{h_1} = W(\boldsymbol{x}_p - \boldsymbol{x}_i)\right)$.
3: **if** «*restrained*» **then**
4: add contribution $\frac{q_i}{h_1^3} W_{ip}^{h_1}$ to the mesh Ω_{h_1}. ▷ Determine $\rho[\mathsf{R}_0]$ - (3.35)
5: **end if**
6: **end for**
7: Determine the electric field $\mathscr{E}_{h_1}$ on Ω_{h_1} ▷ *see* equation (3.40)
8: **for each** particle i **do**
9: Compute and store the forces using $\mathscr{E}_{h_1}$ and $(W_{ip}^{h_1})$. ▷ *see* equation (3.41)
10: **end for**

Algorithm 2 P3M for problem (P_2)

for each particle i **do**
  Determine assignment coefficients on the mesh Ω_{h_2} $\left(W_{ip}^{h_2} = W(\boldsymbol{x}_p - \boldsymbol{x}_i)\right)$.
  if «*active*» **then**
    add contribution $\frac{q_i}{h_2^3} W_{ip}^{h_2}$ to the mesh Ω_{h_1}. ▷ Determine $\rho[\mathrm{A}_0]$ - (3.35)
  end if
end for
Determine the electric field $\mathscr{E}_{h_2}$ on Ω_{h_2} ▷ *see* equation (3.40)
for each particle i **do**
  Retrieve forces associated to the sub-problem (P_0) ▷ *see* equation (3.41)
  Add the forces using $\mathscr{E}_{h_2}$ and $(W_{ip}^{h_2})$. ▷ *see* equation (3.42)
end for

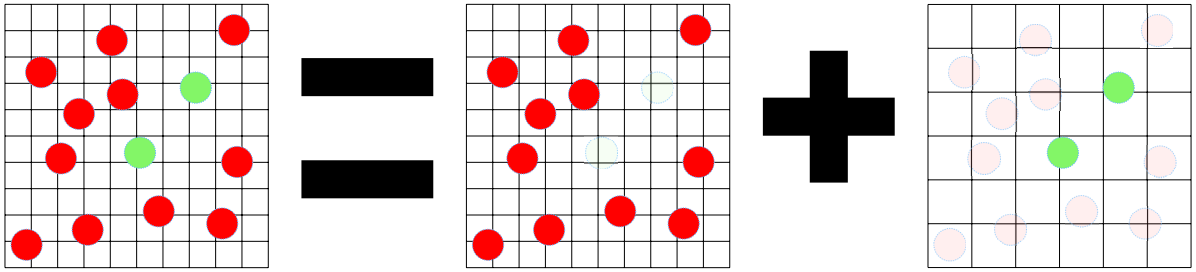


Figure 3.7: Decomposition of P3M between active (green) and restrained (red) particles.

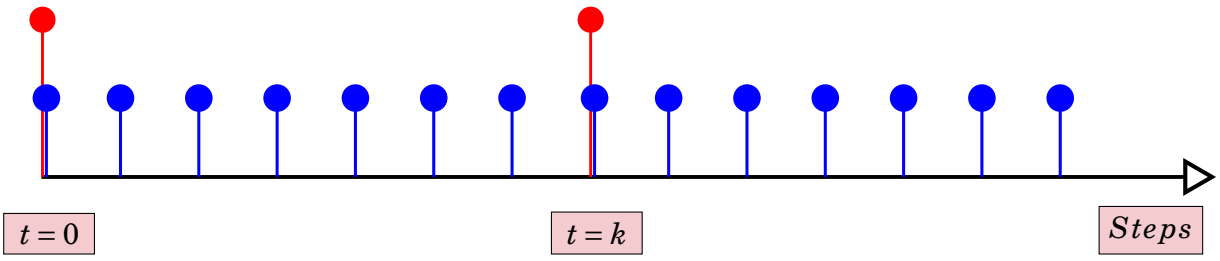


Figure 3.8: Sketch of incremental P3M. Sub-problems (P_1), here in red, and (P_2), in blue, are solved on different timescales.

Incremental evaluation of energy – Another critical point that must be cleared up, is the determination of the energy. Unfortunately, the splitting we are introducing here, doesn't allow a fast (enough) calculation of the energy. In order to retrieve the long range term, it is possible to use equation (3.30). Therefore, one may compute separately the contributions $E^{(1)}$ and $E^{(2)}$ resulting from sub-problems (P_1) and (P_2).

$$E^{long} \simeq E^{(1)} + E^{(2)} \tag{3.43}$$

where

$$E^{(1)} = \frac{1}{2}\sum_{i=1}^{N} q_i \Phi^{(1)}(\boldsymbol{x}_i) \text{ and } E^{(2)} = \frac{1}{2}\sum_{i=1}^{N} q_i \Phi^{(2)}(\boldsymbol{x}_i).$$

Unfortunately, this formulation requires an explicit determination of the potentials $\Phi^{(1)}$ and $\Phi^{(2)}$ induced by sub-problems (P_1) and (P_2). Unlike AD-P3M, $i\boldsymbol{k}$-P3M doest not compute the interaction potentials. Therefore, an incremental update of energy using $i\boldsymbol{k}$-P3M may require at each timestep an additional backward FFT on the coarser grid. Conversely, any extra FFT is needed for an incremental AD-P3M. Furthermore, we want to empathize that $E^{(1)}$ (*resp.* $E^{(2)}$) does not correspond to the energy embedded in the set of restrained (*resp.* active) particles. Therefore, we cannot use the trick employed earlier (*see* remark 3.1) to avoid the explicit evaluation of the interaction potential while computing E^{long}.

In the following sections, we will present in more detail the incremental update of electrostatic forces. We will strive to define the various parameters of the incremental method (grid resolutions, the frequency of calculations on the finer grid). In addition, it will be necessary to characterize and control the error induced by the method. To do so, we will distinguish two specific scenarios. Foremost, we will assume that no particle switches its state throughout the simulation (*see* section 3.5.1). Then we will extend our findings to a more generic case (*see* section 3.5.2).

3.5.1 Case 1 - No switched particles

Under the assumption (A_1), the sub-problem (P_1) is solved once with a $i\boldsymbol{k}$-P3M (Algorithm 1). At each timestep, we want to solve (P_2) with a reduced mesh size. However, we need to justify that the proposed approach doesn't introduce spurious errors. Precisely, it is compulsory to connect, for a given accuracy, the needed grid size to the number of unfrozen particles of the system. Fortunately, for the initial system of N particles, this link is provided by the work from Deserno [53, 54] and Hockney [90].

Let us define the RMS error in forces as

$$\Delta f := \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{f}_i - \boldsymbol{f}_i^{exact}\right)} \tag{3.44}$$

where $\boldsymbol{f}_i$ is the calculated force on particle i and $\boldsymbol{f}_i^{exact}$ the exact force. Using the mean square force error for two unit charges χ, an initial estimation of the RMS force error is provided by [53].

$$\Delta f \simeq \frac{Q^2}{\sqrt{N}}\chi. \tag{3.45}$$

This estimation links the RMS force error with the number of charges present in the system (N), the total square sum of charges Q^2 (*see* equation (3.12)) and the measure of the error involved in the chosen method (χ). Hockney and Eastwood extends this estimation [90]

$$\Delta f \simeq \sqrt{\frac{\mathscr{F}}{NL^3}}Q^2. \tag{3.46}$$

where, $\mathscr{F} = \chi^2 L^3$ is the displacement-averaged total squared deviation of $\boldsymbol{F}(\boldsymbol{x}_2 - \boldsymbol{x}_1, \boldsymbol{x}_1)$, the mesh-based approximation of the k-space contribution of the force between two unit charges located at positions $\boldsymbol{x}_1$, $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{x}$, from a reference inter-particle force $\boldsymbol{F}_r(\boldsymbol{x}_2 - \boldsymbol{x}_1)$. Basically,

$$\mathscr{F} := \frac{1}{h^3}\int_{h^3} \mathrm{d}^3 x_1 \int_{L^3} \mathrm{d}^3 x [\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{x}_1) - \boldsymbol{F}_r(\boldsymbol{x})]^2. \tag{3.47}$$

Here, the inner integral accumulates the error over all particle separations whilst the outer integral over averages over all possible locations of the first particle within a mesh cell. The idea behinds this formula comes from the fact that forces between two particles should depend on their separation $\boldsymbol{x}$ only. However, since we are using grids, the computed force also depends on the positions of the particles relative to the mesh.

Given a charge assignment function, $\mathscr{F}$ is a functional of the influence function. $\mathscr{F}$ is minimized by G_{opt}. Consequently, $\mathscr{F}[G_{opt}]$ gives the *optimal error* of the method. The expression of this error is given in appendix A.2, in case of $i\boldsymbol{k}$-P3M. Nonetheless, for the sake of simplicity, we restrict ourselves to the analytical approximation of $\mathscr{F}[G_{opt}]$ derived by Deserno and Holm [53], namely,

$$\Delta f \simeq \frac{Q^2 (h\beta)^P}{\sqrt{N} L^2} \sqrt{\sqrt{2}\pi\beta L \sum_{m=0}^{P-1} a_m^{(P)} (h\beta)^m} \tag{3.48}$$

where $\beta = \dfrac{1}{\sqrt{2}\sigma}$ and $a_m^{(P)}$ denote expansion coefficients given in appendix A.3. We recall that h is the mesh spacing and P the order of the assignment function. This approximation is valid in the limit of small $h\beta$. Therefore, at the first order,

$$\Delta f \propto \frac{Q^2 h^P}{\sqrt{N}} \tag{3.49}$$

Let us consider the sub-problem (P_2) that is dedicated to the system of N_a active particles. In appendix A.1, we extend the approximation (3.45) in order to estimate Δf_2, the RMS error in the forces induced by active particles:

$$\Delta f_2 \simeq \frac{Q_a^2}{\sqrt{N_a}}\chi \tag{3.50}$$

Similarly to equation (3.12), Q_a^2 designates the square sum of active charges. Therefore, from equation (3.49),

$$\Delta f_2 \propto \frac{Q_a^2 h_a^P}{\sqrt{N_a}} \tag{3.51}$$

In order to control the accuracy of the computation associated to (P_2), we define γ as the ratio between Δf_2 and Δf. The estimations (3.49) and (3.51) leads to

$$\gamma = \frac{\Delta f_2}{\Delta f} \simeq \frac{Q_a^2 h_a^P}{Q^2 h^P}\sqrt{\frac{N}{N_a}} \tag{3.52}$$

Thus, we can estimate the grid size (per dimension) of problem (P_2) $M_2 = \dfrac{1}{h_a}$ as follows

$$M_2 \simeq M\left[\frac{1}{\gamma}\frac{Q_a^2}{Q^2}\sqrt{\frac{N}{N_a}}\right]^{\frac{1}{P}} \tag{3.53}$$

At this point, let us assume that the system is homogeneous. Thus, an estimation of the ratio of the sum of squared valences is

$$\frac{Q_a^2}{Q^2} \simeq \frac{N_a}{N}. \tag{3.54}$$

Therefore, equation (3.53) can be written as

$$M_2 \simeq M\left[\frac{1}{\gamma}\sqrt{\frac{N_a}{N}}\right]^{\frac{1}{P}} \tag{3.55}$$

$$= M\left[\frac{1}{\gamma}\sqrt{\alpha_a}\right]^{\frac{1}{P}} \tag{3.56}$$

When enough particles are restrained, the sub-problem (P_2) can be solved with a prescribed accuracy on a reduced mesh size Ω_{h_2}. In figure 3.9, is shown the ratio $\dfrac{M^3}{M_2^3}$

between the number of grid points of the meshes Ω_{h_2} and Ω_h. When $\gamma = 1$ and $P = 5$, the grid size is reduced by a factor 2 for systems which contain 10% of active particles. In practice, γ cannot be set at 1. In fact, as computed forces of (P_2) have to be summed up with those originating from (P_1), our estimation of the RMS errors in forces has to take account of errors induced by both (P_1) and (P_2). We may assume that both error sources are uncorrelated[2] and thus

$$(\Delta f_{1+2})^2 = (\Delta f_1)^2 + (\Delta f_2)^2 \tag{3.57}$$

Here Δf_{1+2} is the RMS error in forces resulting from the splitting strategy. Assuming that the error levels are equivalent, we can expect that

$$(\Delta f_{1+2})^2 \simeq 2(\Delta f_2)^2 . \tag{3.58}$$

Finally, since our strategy has to maintain the order of accuracy of a traditional ik-P3M

$$(\Delta f) \simeq (\Delta f_{1+2}) \implies \gamma = \frac{\sqrt{2}}{2}. \tag{3.59}$$

In figure 3.9, we plotted the ratio of grid sizes using $\gamma = \frac{\sqrt{2}}{2}$ and $P = 5$. When 10% of particles are active the ratio $\frac{M^3}{M_2^3}$ is equal to 1.62. However, our approach is no longer rewarding when less than 50% of particles are restrained.

Moreover, the mesh associated to the problem (P_1) has to account of this modification. The derivation (3.55) can be extended to restrained particles. For instance, when 90% of particles are restrained the resolution of sub-problem (P_1) requires $\sim 1.19M^3$ grid points.

With that in mind, let us imagine that the initial problem has to be solved on a $100 \times 100 \times 100$ grid. In addition, the system contains 10% of active particles. From the above, (P_1) has to be solved on a $106 \times 106 \times 106$ mesh whereas (P_2) requires a $85 \times 85 \times 85$ grid. Although (P_1) is penalized, by solving only (P_2) at each timestep, one reduces the cost of the 4 FFTs needed in ik-P3M. In principle, the penalty can be paid once during the initialization phase. Furthermore, it is possible to configure Ω_{h_1} such that it can hold all particles, namely,

$$M_1 = M\gamma^{-1/P} \tag{3.60}$$

[2] or statistically independent

The price to pay is negligible since this grid is seldom used. For instance, here the ratio of the numbers of grid points reads $M_1^3/M^3 \simeq 1.23$ when $P = 5$.

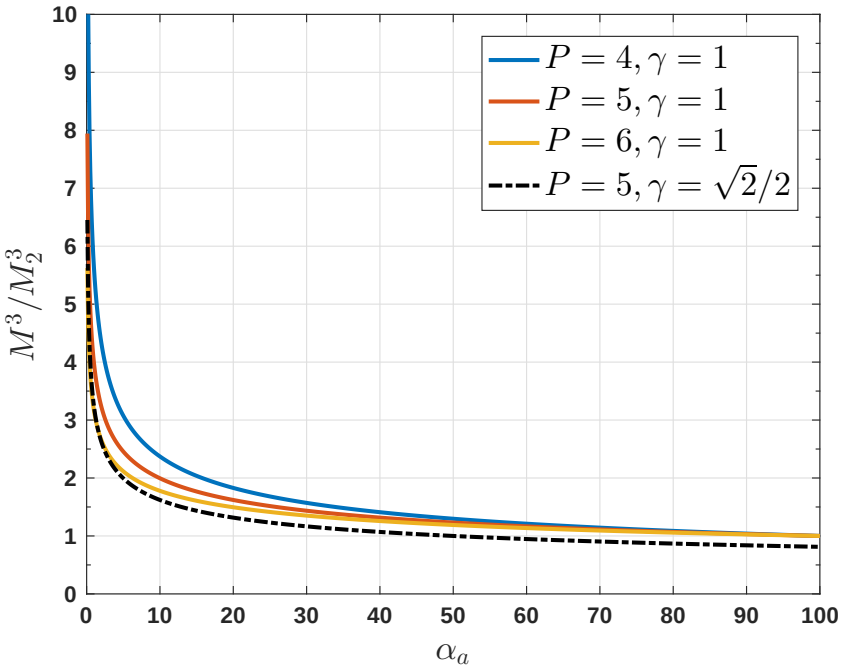


Figure 3.9: Ratio of numbers of grid points between the original mesh and the grid associated to the problem (P_2)

Validation – In order to validate our assumptions, we applied our splitting strategy to the system of SPC/E water (36000 particles). Some randomly selected particles are restrained. Various ratios of restrained/active particles were evaluated. For each configuration, forces were computed with a highly accurate ewald summation $\left(\frac{\Delta f}{|f|} \simeq 10^{-10}\right)$, a traditional P3M with a prescribed accuracy $\left(\frac{\Delta f}{|f|} \simeq \epsilon\right)$. Here, we parameterized both methods with the relative RMS error in forces, namely $\frac{\Delta f}{|f|}$, since this term can be estimated from the magnitude of this error for two unit charges (χ) [112].

Ewald forces are considered as exact forces while classical ik-P3M calculations are set as a reference for a given precision. For a prescribed ϵ, P3M is tuned in order to achieve a target RMS force error Δf^{target} using the formula (3.44). Δf^{target} is used to parameterize the grid size of P3M and the ewald splitting parameter. For P3M calculations the real space cutoff was set at 9.8Å and particles were mapped to the grid with a 5^{th} order assignment function.

Then, we tuned the incremental P3M such that the relative accuracy ϵ is also achieved.

As explained, two meshes (Ω_{h_1}, Ω_{h_2}) have been used. Both are set up with an expected RMS error equals $\gamma\Delta f^{target}$. Ω_{h_1} holds forces induced by restrained particles while Ω_{h_1} is dedicated to the electrostatic quantities induced by active particles. In order to maximize the efficiency of FFT calculations, we only use composite grid sizes (*see* figure 2.2).

Moreover, table 3.2 prints the parameters used in our benchmark. In this table, are shown the effective RMS errors associated to incremental P3M (Δf_{1+2}) and the classic ik-P3M (Δf). In addition, we denote by S_{long} the speedup achieved for P3M long-range calculations, namely,

$$S_{long} = \frac{T_{long}}{T_{long}^{(1+2)}} \tag{3.61}$$

where T_{long} is the CPU time needed in ik-P3M long-range calculations and $T_{long}^{(1+2)}$ is the CPU time needed in incremental P3M.

The incremental P3M seems to be efficient for high accuracy calculations where large grid sizes are used. In figure 3.10, the method showed a good preservation of the accuracy while reducing FFT calculations (Table 3.2). For instance, when the relative rms error ϵ is equal to 10^{-6} and when 20%, long-range computations were accelerated by a factor 1.5. The short-range calculations can be sped up as well. For the configuration, the total speedup (including short-range calculations) was 2.1.

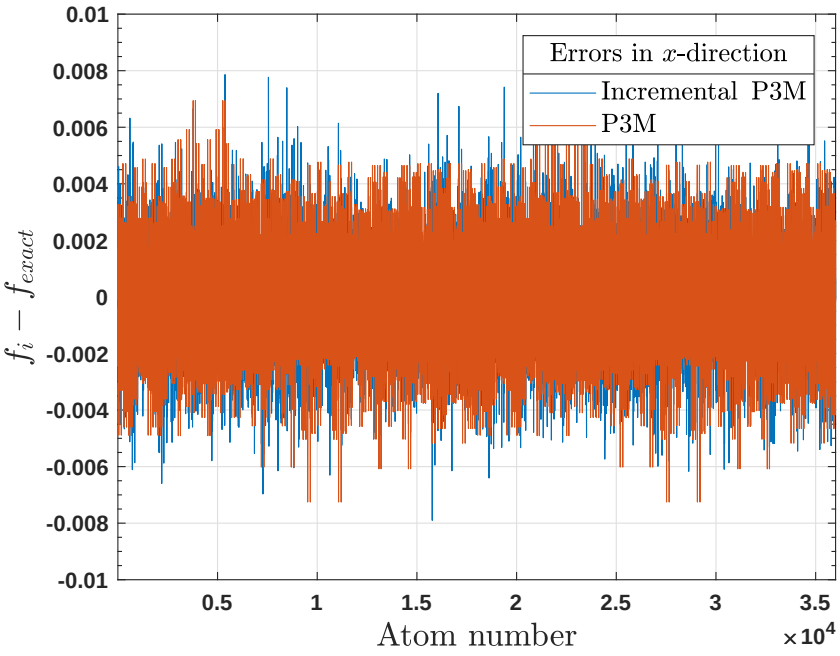


Figure 3.10: Error in forces (x-direction) of incremental P3M .

α_a	M	M_1	M_2	Δf	Δf_{1+2}	S_{long}
$\epsilon \simeq 10^{-4}$ and $\Delta f^{target} = 3.8 \times 10^{-2}$						
1%	36	40	30	2.50×10^{-2}	1.90×10^{-2}	×1.9
5%	36	40	30	2.50×10^{-2}	2.32×10^{-2}	×1.8
10%	36	40	32	2.50×10^{-2}	2.92×10^{-2}	×1.7
20%	36	40	36	2.50×10^{-2}	2.05×10^{-2}	×1.1
40%	36	40	36	2.50×10^{-2}	2.19×10^{-2}	×1.0
$\epsilon \simeq 10^{-5}$ and $\Delta f^{target} = 3.9 \times 10^{-3}$						
1%	60	64	45	2.85×10^{-3}	3.01×10^{-3}	×2.2
5%	60	64	50	2.85×10^{-3}	2.82×10^{-3}	×1.8
10%	60	64	54	2.85×10^{-3}	2.79×10^{-3}	×1.34
20%	60	64	60	2.85×10^{-3}	2.46×10^{-3}	×1.01
40%	60	64	60	2.85×10^{-3}	2.70×10^{-3}	×1.0
$\epsilon \simeq 10^{-6}$ and $\Delta f^{target} = 3.4 \times 10^{-4}$						
1%	100	108	72	2.96×10^{-4}	2.86×10^{-4}	×3.1
5%	100	108	80	2.96×10^{-4}	2.87×10^{-4}	×2.2
10%	100	108	81	2.96×10^{-4}	2.85×10^{-4}	×1.9
20%	100	108	90	2.96×10^{-4}	2.89×10^{-4}	×1.5
40%	100	108	90	2.96×10^{-4}	2.95×10^{-4}	×1.4
$\epsilon \simeq 10^{-7}$ and $\Delta f^{target} = 3.8 \times 10^{-5}$						
1%	180	180	120	2.37×10^{-5}	2.83×10^{-5}	×3.1
5%	180	180	135	2.37×10^{-5}	2.99×10^{-5}	×2.5
10%	180	180	144	2.37×10^{-5}	3.02×10^{-5}	×2.3
20%	180	180	150	2.37×10^{-5}	3.00×10^{-5}	×1.8
40%	180	180	160	2.37×10^{-5}	2.95×10^{-5}	×1.5

Table 3.2: Performance of incremental P3M on the SPC/E system.

3.5.2 Case 2 - Particles can switch state

Hitherto, we assume that at each timestep, the sets R_k and A_k are constant ($\mathsf{R}_k = \mathsf{R}_0$ and $\mathsf{A}_k = \mathsf{A}_0$). However, particles may change their state over the course of ARMD simulations. Moreover, it is necessary to not change (often) the various grid resolutions. In fact, a modification of the grid size requires the re-computation of the influence function G_{opt} and the reconfiguration of the FFT solver. The cost of these operations can be important but negligible in the traditional P3M – since these operations occur at the initialization stage —. For the problem (P_1), this issue is resolved by configuring Ω_{h_1} as prescribed by equation (3.60).

Another crucial aspect we need to ensure is the fact that the summation of the solutions of the sub-problems (P_1) and (P_2) gives the desired quantity up to a prescribed accuracy. On the other hand, we still want to be able to solve the sub-problem P_1 once or at least less frequently. Therefore, we need to understand how switched particles affect the problem's decomposition (3.42).

At this point, we want to retrieve as much information as possible from a computed solution of (P_1). Assuming that the sub-problem (P_1) has been computed at $t = 0$, it is likely that at $t = k$, some particles of R_0 have not changed their location. Those particles which have never been active between timesteps $t = 0$ and $t = k$, are stored in a subset R_0^{k+} of R_0. Analogously, we denote by R_0^{k-} the subset of particles of R_0 that have switched, and thus been displaced, at least once between $t = 0$ and $t = k$. From these definitions,

$$\forall k \in \mathbb{N}, \quad \mathsf{R}_0 = \mathsf{R}_0^{k+} \cup \mathsf{R}_0^{k-} \text{ and } \mathsf{R}_0^{k+} \cup \mathsf{R}_0^{k-} = \emptyset \tag{3.62}$$

The precomputed solution of (P_1) embeds, at each timestep, the contribution of particles in R_0^{k+}. When R_0^{k+} is large enough, it might be useful to reuse the solution of (P_1) computed at $t = 0$. Therefore, it is compulsory to determine the magnitude of R_0^{k+} which corresponds to a measure of the valuable information embedded in the solution of the sub-problem (P_1). A correction of the solution of (P_2) is required in order to retrieve the true long-ranged contributions. This correction (P_2) must take into account the fact that particles in R_0^{k-} has been displaced and their contribution (from (P_1)) was computed with wrong coordinates.

In order to define properly the needed correction, we need to examine the flux of particles that move from the subset $\mathsf{R}_0^{(k-1)+}$ to R_0^{k-}. This flux is essentially related to the probability that a given restrained particle switches to active state. This probability is denoted by $\mu = \Pr(\mathsf{R} \to \mathsf{A})$. Similarly, $\Pr(\mathsf{A} \to \mathsf{R})$ defines the probability that an active particle switches to restrained state. We can link μ to the ratios α_r (of restrained particles)

and α_s (of switched particles). In practice, these ratios are functions of time that converge to an equilibrium value. Let us assume that the equilibrium state is reached. For the sake of simplicity, these ratios are taken constant and equal to their equilibrium values. It is straightforward that

$$\alpha_r \Pr(\mathsf{R} \to \mathsf{A}) = \alpha_a \Pr(\mathsf{A} \to \mathsf{R}) = \frac{\alpha_s}{2} \tag{3.63}$$

Hence,

$$\mu = \Pr(\mathsf{R} \to \mathsf{A}) = \frac{\alpha_s}{2\alpha_r} \tag{3.64}$$

Between two successive timesteps $k-1$ and k, we can assume that $\mu \left|\mathsf{R}_0^{(k-1)+}\right|$ particles from the subset $\mathsf{R}_0^{(k-1)+}$ have switched to active. Thus,

$$\left|\mathsf{R}_0^{k+}\right| = (1-\mu)\left|\mathsf{R}_0^{(k-1)+}\right| \tag{3.65}$$

This geometric progression gives

$$\begin{aligned} \left|\mathsf{R}_0^{k+}\right| &= (1-\mu)^k \left|\mathsf{R}_0^{0+}\right| \\ &= (1-\mu)^k \left|\mathsf{R}_0\right| \end{aligned} \tag{3.66}$$

Moreover, at the timestep $t = k$, the solution of (P_1) carries unwanted contributions emanating from particles in R_0^{k-}. For a given particle p in R_0^{k-}, we must remove its contribution which has been computed with (P_1) at $t = 0$. To do so, we define the virtual particle $\bar{p}$ located at the coordinates of p but with an opposite charge:

$$\forall p \in \mathsf{R}_0^{k-}, \quad \left\{ \begin{array}{lcl} \boldsymbol{x}_{\bar{p}}^{(t=k)} & := & \boldsymbol{x}_p^{(t=0)} \\ q_{\bar{p}} & := & -q_p \end{array} \right. \tag{3.67}$$

These virtual particles can be stored in the collection V_k whose size reads.

$$|\mathsf{V}_k| = \left|\mathsf{R}_0^{k-}\right| = \left[1 - (1-\mu)^k\right] |\mathsf{R}_0| \tag{3.68}$$

Let us denote $\rho[\mathsf{V}_k]$ as the charge density associated to the virtual particles defined in V_k. Likewise, $\rho\left[\mathsf{R}_0^{k-}\right]$ denotes the charge density induced by particles from R_0 that have been active at least once since $t = 0$ (using their current coordinates). We claim that, at $t = k$, the charge density corresponding to the current distribution of particles is given by

$$\rho[t=k] = \underbrace{\rho_1[t=0]}_{\text{Problem } (P_1)} + \underbrace{\rho[\mathsf{A}_0] + \rho[\mathsf{V}_k] + \rho\left[\mathsf{R}_0^{k-}\right]}_{\text{Problem } (P_2)} \tag{3.69}$$

Here, $\rho_1[t=0] = \rho[\mathsf{R}_0]$ at timestep $t=0$. The long-ranged contributions (forces or electric field) stemming from $\rho[\mathsf{R}_0]$ can be computed at $t=0$ with a given mesh Ω_{h_1}. At each timestep, the missing contribution is solved on a separate mesh Ω_{h_2} and then added to the save. Ω_{h_2} will be precisely characterized if we are able to determine N_2, the number of particles involved in the definition of the charge density

$$\rho_2[t=k] = \rho[\mathsf{A}_0\,] + \rho[\mathsf{V}_k\,] + \rho\left[\mathsf{R}_0^{k-}\,\right].$$

At each timestep, an estimation of N_2 reads

$$N_2(k) = |\mathsf{A}_0\,| + |\mathsf{V}_k\,| + \left|\mathsf{R}_0^{k-}\right| \tag{3.70}$$

$$= \alpha_a N + 2\left[1-(1-\mu)^k\right]|\mathsf{R}_0| \tag{3.71}$$

For a prescribed accuracy, one can determine $M_2(k)$ the number of grid points (per dimension) required for Ω_{h_2} in order to maintain the accuracy after k timesteps (*see* equation (3.53)). For instance, under the assumption that the system is homogeneous,

$$M_2(k) \simeq M\left[\frac{1}{\gamma}\sqrt{\alpha_2(k)}\right]^{\frac{1}{P}} \tag{3.72}$$

where

$$\alpha_2(k) := \frac{N_2(k)}{N}$$

$$= \alpha_a + 2\left[1-(1-\mu)^k\right]\alpha_r \tag{3.73}$$

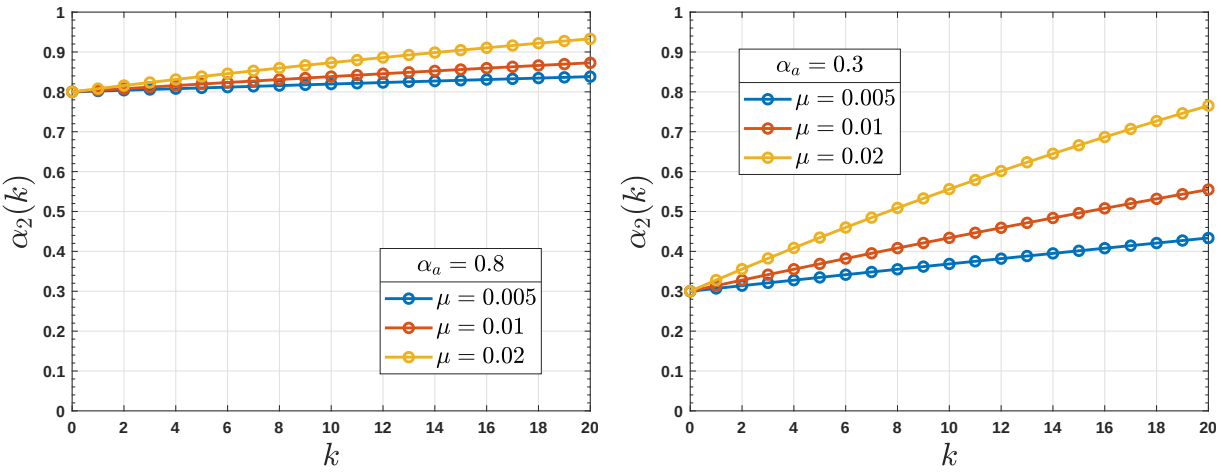
$$= 1 + \left[1-2(1-\mu)^k\right]\alpha_r. \tag{3.74}$$

Figure 3.11 outputs the evolution of $\alpha_2(k)$ as a function of the timestep $t=k$ when the proportion of restrained particles $\alpha_r \in \{0.2, 0.7\}$.

For $\alpha_r = 0.2$ ($\alpha_r = 0.8$) and $\mu = 0.005$, if we want to maintain the prescribed precision up to $t=20$, we must configure Ω_{h_2} such that it may hold up to 84% of particles (including virtual particles). In this case, the difference with the ideal case where no particle switches is rather small. For the same value of α_r, an increase of the proportion of switched particles will have a limited impact on the grid size of Ω_{h_2} (*e.g.* when $\mu = 0.01$ $\alpha_2 - \alpha$ does not exceed 8% at $t=20$).

Conversely, μ has a significant impact on $\alpha_2(k)$ when a large amount of particles is restrained. For instance, when $\alpha_r = 70\%$ and $\mu = 0.005$, $\alpha_2 - \alpha$ exceeds 10% for $t \geq 15$. Meanwhile $\mu = 0.02 \implies \alpha_2(20) > 2\alpha_a$ which leads to a very fine grid for the resolution of (P_2). Although $\mu = 0.02$ is uncommon in ARMD simulations, this example points out the

fact the performance of incremental P3M can be limited by the ratio of switched particles. In practice, μ rarely exceeds 0.5% in ARMD simulations (*e.g.* table 1.2).



(1) $\alpha_r = 0.2$ (2) $\alpha_r = 0.7$

Figure 3.11: Evolution of $\alpha_2(k)$ as a function of the timestep k. Here $\mu = \dfrac{\alpha_s}{2\alpha_r}$ denotes the probability that a restrained particle switches to active.

3.5.3 Optimal parameters

The previous section was dedicated to the control of the grids required by the incremental P3M. Basically, we showed that the resolution of the coarsest grid depends on the proportion of restrained particles and the ratio between the number of switched particles and the number of restrained particles. At this point, it becomes critical to be able to control how frequently we should employ the finest mesh Ω_{h_1}. In other words, we should determine when the incremental P3M must be reset.

Let us assume that every $k+1$ steps, computations occur on the finest grid Ω_{h_1}. Meanwhile (P_2) must be solved at every timestep. At each timestep, the cost of the computations can be estimated, at the first order, with the number of grid points involved. After k steps, the total cost of the incremental P3M reads $c_1(k) = M_1^3 + (k+1)M_2^3$. Conversely, if one utilizes a classic P3M, the computation cost is equal to $c_0(k) = (k+1)M^3$.

Let us define the «*grid efficiency*» as the ratio between the total grid points operations involved in the classical P3M and those involved in the incremental P3M.

$$\eta(k) := \frac{c_0(k)}{c_1(k)} = \left(\frac{a}{k+1} + \frac{M_2^3}{M^3} \right)^{-1} \tag{3.75}$$

where $a = \dfrac{M_1^3}{M^3}$. Using (3.72), the grid efficiency can be rewritten as:

$$\eta(k) = \left[\frac{a}{k+1} + \left(\frac{1}{\gamma} \sqrt{\alpha_2(k)} \right) \frac{1}{P} \right]^{-1} \tag{3.76}$$

$$= \left[\frac{a}{k+1} + \left(\frac{1}{\gamma} \sqrt{1 + \left(1 - 2(1-\mu)^k\right) \alpha_r} \right) \frac{1}{P} \right]^{-1}$$

It is possible to estimate the optimal timestep k_{opt} by finding the maximum of the grid efficiency. Figure 3.12 shows the evolution of the grid efficiency as a function of timestep for $P = 5$ and $\alpha_r = 90\%$. In the ideal case where no particle switches its state ($\mu = 0$), the optimal timestep is indeed $k_{opt} = +\infty$. In this example, the ideal grid efficiency corresponds to ~ 1.75.

Conversely, if particles are allowed to switch at the rate $\mu = 0.5\%$, $\eta(k)$ is maximal when $k_{opt} = 11$. In order to preserve the accuracy of the computations up to $t = k_{opt}$, the mesh size of Ω_{h_2} is adjusted to $M_2(k_{opt})$. As a result, the grid efficiency decreases to ~ 1.25 .

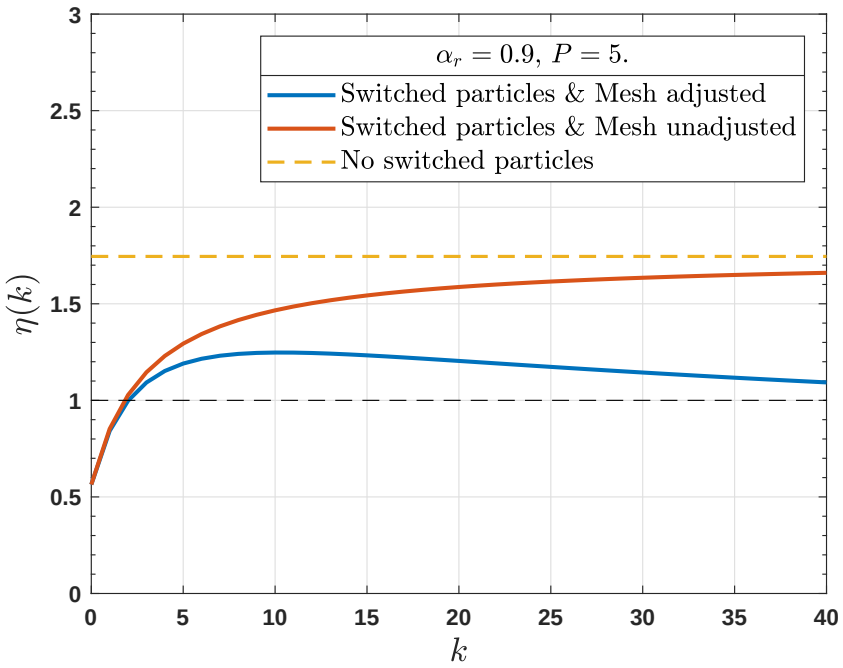


Figure 3.12: Evolution of the grid efficiency $\eta(k)$. $\eta(k)$ is plotted in red for $\mu = 0.5\%$ and (P_2) is solved on a grid whose size is adjusted to a grid size to $M_2(k)$. In blue, $\mu = 0.5\%$ but the sub-problem (P_2) is solved with a grid size $M_2(0)$. The ideal case where $\mu = 0.0$ is represented with the orange dashed line.

Remark 3.2. *Let assume that we do not adjust the grid size of Ω_{h_2} to $M_2(k)$. Instead, we use $M_2(0)$ as grid size. Consequently, the grid efficiency $\eta(k)$ increases and its value at $k = k_{opt}$ is about* 1.5. *However, this approach implies that the level of accuracy is not guaranteed throughout the simulation (*see *figure 3.13).*

Figure 3.13 shows the evolution of $\frac{\Delta f}{|f|}$, the relative RMS force error, throughout simulations of the system of SPC/E water . During these simulations, 90% of particles are restrained at each timestep. In addition, we allow 0.9% of particles to switch to active/restrained ($\mu = 0.005$). A reference simulation was performed on a 100^3 grid with the traditional ik-P3M. The real space cutoff was again set at 9.8Å and the achieved relative RMS force error was $\sim 2.6 \times 10^{-6}$. Then, we applied incremental P3M on the water system and compared the ability of both methods to determine forces. The sub-problem P_2 was solved with a 90^3 grid which allowed us to maintain, for 12 steps, Δf below the desired threshold. Thus, we reset the algorithm by recomputing the solution of (P_1). After 100 timesteps, the long-range computations were accelerated by a factor 1.35. Meanwhile the short-range has benefited of a $\times 3.9$ speedup.

Moreover, while using a $81 \times 81 \times 81$ mesh for (P_2), the target accuracy was achieved only for the first step. In counterparts, long-range calculations were performed 1.6 times faster. Nevertheless, the error level increases up to 3.7×10^{-6} after 12 timesteps.

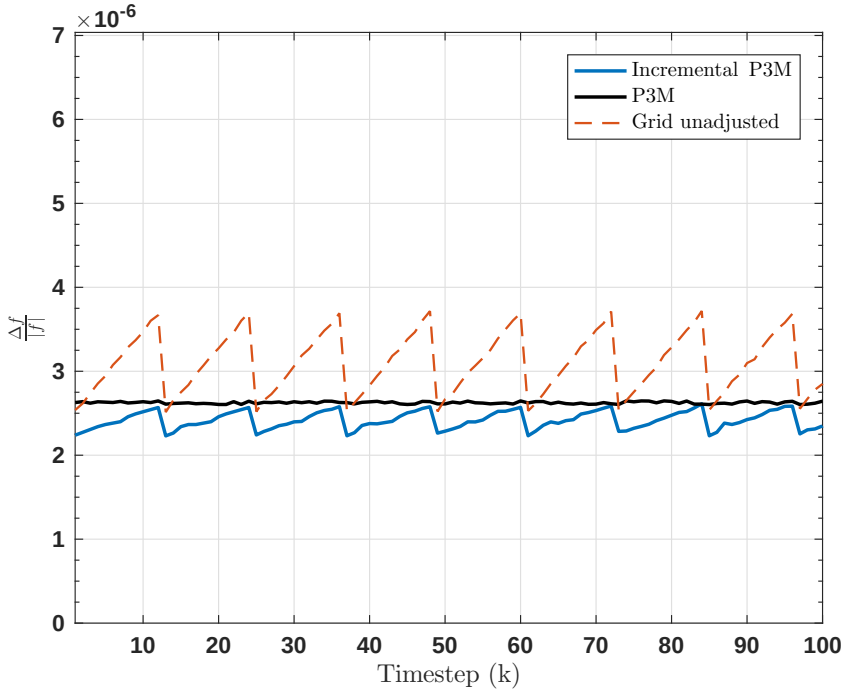


Figure 3.13: Evolution of the RMS force error in incremental P3M

3.6 Conclusion

In order to speed up the calculation of electrostatic quantities in ARMD simulations, we have proposed various improvements to Fourier based methods. In section 3.3, real- and Fourier space calculations of Ewald summation has been accelerated while increasing the number of restrained particles. In practice, the performance of the incremental evaluation of Fourier space contributions is lower than the performance obtained with the real-space part. Nonetheless, we achieved a significant speedup with respect to the classical Ewald method, especially for the computation of the electrostatic potential energy. Unfortunately, the resulting acceleration does not allow the incremental Ewald method to outperform particle mesh methods such as P3M.

Therefore, we extend our analysis to mesh-based Ewald methods. Precisely, we propose in section 3.5 a new algorithm based on P3M. The suggested approach can be extended to any mesh-based Ewald technique. Moreover, the method takes advantage of the real space acceleration proposed for Ewald summation. The Fourier space part was decomposed into two contributions for the ARMD systems which are determined using different scales of space and time. The first contribution is sparingly computed (and stored) using a fine grid while the remaining contribution is evaluated at each timestep with a coarse grid. The desired electrostatic quantity is obtained by summing the stored and the computed terms. The stored contribution is recomputed every k timesteps in order to maintain the simulation under prescribed error magnitude. An optimal value of k is estimated by maximizing (3.76). Although, the performance of the proposed strategy is limited by the amount of particles that switch state each timestep, we showed that the incremental P3M can reduce the cost of electrostatics quantities.

Nonetheless, some aspects of the incremental method can be criticized. First, the determination of the optimal timestep requires a good knowledge of α_s, the ratio of switched particles. In addition, a small amount of switched particles is needed for a valuable speedup of the long-range calculations in P3M. When more than half of particles are active, only the real space calculations can be accelerated. Moreover, the optimal size of the coarser mesh can be inadequate for FFT calculations (*e.g* prime numbers). In particular, the incremental update of electrostatics energy in ik-P3M is strongly limited. Finally, the incremental update of the Fourier space energy is not ideal for $i\boldsymbol{k}$-differentiated particle mesh methods.

Although some limitations can be bypassed (for instance, by using the incremental P3M when α_s reached its equilibrium value or by broadening slightly the coarser mesh

size or by favoring real space calculations), we want to derive a more efficient method which requires less tweaks.

CHAPTER 4

INCREMENTAL MESHED CONTINUUM METHOD

Contents

4.1 Overview

In almost all fields of physics and engineering sciences, large linear/nonlinear systems have to be solved . Usually, these systems emanate from the discretization of PDEs that describe a wide variety of phenomena such as heat, sound, quantum mechanics, elasticity, fluid dynamics, or electrostatics [82, 209]. The discretization of elliptic PDEs leads to sparse and symmetric positive definite matrices. Iterative algorithms methods, such as multigrid, are among the fastest numerical methods for solving this kind of equations [96]. Multigrid solvers can reach an asymptotically optimal $\mathcal{O}(N)$ complexity, where N is the number of unknowns in the corresponding linear system. Multigrid methods are applicable in a variety of fields. Hence, this topic yields an impressive and constantly growing body of literature and research.

In MD simulations, iterative methods were often employed in order to reduce the complexity of calculation of the electrostatic component of atomic forces [85, 180, 206]. In 2008, MCM was introduced by Bolten and Sutmann in order to efficiently compute electrostatic potential. They exploited the fact that this potential is strongly connected to the Poisson equation ($\Delta\phi = f$).

Foremost, let us recall that we want to compute the electrostatic potential Φ generated by N charged particles with charge q_i at position $\boldsymbol{x}_i$ in a cubic box $\mathscr{B}$ of length L. To ensure periodic boundary conditions (p.b.c), we assume that the simulation box $\mathscr{B}$ is replicated an infinite number of times [185] :

$$\Phi(\boldsymbol{x}) = -\frac{1}{4\pi\epsilon_0} \sum_{\boldsymbol{n}\in\mathbb{Z}^3} \sum_{\substack{i=1 \\ \boldsymbol{x}_i^0 \neq \boldsymbol{x}}}^{N} q_i \frac{1}{||\boldsymbol{x} - \boldsymbol{x}_i^{\boldsymbol{n}}||}. \tag{4.1}$$

Here, $\boldsymbol{x}_i^{\boldsymbol{n}} = \boldsymbol{x}_i + \boldsymbol{n}L, \boldsymbol{n} \in \mathbb{Z}^3$ are the replicated particle images, and ϵ_0 is the dielectric permittivity. We recall that this potential can be seen as a solution of the following Poisson equation with respect to periodic boundary conditions

$$\Delta\Phi(\boldsymbol{x}) = -\frac{1}{\epsilon_0} \sum_{i=1}^{N} q_i \delta(||\boldsymbol{x} - \boldsymbol{x}_i||) \tag{4.2}$$

where the right-hand side (r.h.s) is described by Dirac δ functions associated to the point charges in the simulation box.

In MCM, the crux of the matter is to solve a modified version of the Poisson equation (4.2). The original expression of equation (4.2) cannot be solved numerically due to discontinuities introduced by Dirac δ functions [10]. Instead of modeling the particles by point charges which can be represented by Dirac impulses, a *smooth* charge density is

used and sampled on a mesh. The density smears the charge of the particle over several grid points. The r.h.s. of the Poisson problem is obtained by sampling the contribution of each particle on the mesh. Therefore, the problem can be solved by a fast Poisson solver, in this case, a traditional multigrid algorithm. After the solution is available on the mesh, the electrostatic quantities of interest can be obtained from this discrete solution by interpolating it back to the particles and applying a near-field correction scheme [25].

In summary, MCM consists in five main phases :

- Evaluation of the **right-hand side** of the Poisson equation
- Computation of the **near-field correction** step
- Resolution of the Poisson equation via a **multigrid method**
- **Interpolation** of the solution at particles positions
- **Computation of the potential and forces** by combining the near-field correction term and the interpolated solution.

Although MCM exploits the optimality of the multigrid approach ($\mathcal{O}(N)$), it has been demonstrated that the method is slower than FFT-based methods such as P3M [10]. The evaluation of the r.h.s. and the near-field correction are the most CPU demanding stages of MCM. Nevertheless, in this chapter, we will show that these steps can be efficiently accelerated with incremental computations. First, we will present and evaluate various aspects of the original method. For instance, the choice of the charge density will be discussed. We refer to section 2.5 for details on the chosen multigrid method. Then, we propose an Incremental Meshed Continuum Method (IMCM) that legitimates the use of this multigrid-based method for the computation of electrostatic quantities in ARMD simulations.

4.2 Meshed Continuum Method

4.2.1 The right-hand side

In section 4.1, Dirac functions are replaced by smooth charge densities. Here we choose point symmetric distributions as charge densities (*see* definition 2.1). Therefore, δ function associated to each particle is replaced by

$$\varphi_i^{\boldsymbol{n}}(\boldsymbol{x}) = \rho_{r_c}(||\boldsymbol{x} - \boldsymbol{x}_i^{\boldsymbol{n}}||). \tag{4.3}$$

Here, ρ_{r_c} is defined as a normalized radially symmetric distribution such that

$$\rho_{r_c}(||\boldsymbol{r}||) = 0 \qquad \text{If } ||\boldsymbol{r}|| > r_c.$$

The normalization guarantees that the charge induced by $q_i\varphi_i^{\boldsymbol{n}}$ is equal to the charge q_i. Furthermore, the compactness implies that the potential induced by the difference in charge distributions $q_i(\varphi_i^{\boldsymbol{n}} - \delta)$ is nil outside of the support of $\varphi_i^{\boldsymbol{n}}$ (*see* lemma 2.1, [78]).

Nonetheless, the optimal choice for the charge distribution is not clear. However, Bolten [25] proposed a centered quadratic B-spline as charge distribution. Moreover, a smooth charge density can be constructed with the help of cardinal B-splines. Precisely, a cardinal B-spline of order $m \in \mathbb{N}^*$ is a piecewise real function of class $\mathscr{C}^{m-2}$ with a limited support $[0, m]$. In practice, ρ_{r_c} can be constructed by centering, rescaling and normalizing a B-spline of a given order (See Section 2.3). In addition, this B-spline can be described with m different polynomials of degree $m-1$[44, 45]. A practical algorithm for calculating the coefficients of polynomials, which determine a cardinal B-spline, was proposed by Milovanović and Udovičić [145]. In section 2.3, we provide some examples of charge distributions.

4.2.2 The near-field correction

Given a smooth charge density, the corresponding potential ϕ can be written as the convolution of ρ_{r_c} with the 3D Green's function :

$$\phi(\boldsymbol{x}) = \int_{\mathbb{R}^3} \frac{\rho_{r_c}(||\boldsymbol{y}||)}{4\pi||\boldsymbol{x}-\boldsymbol{y}||}\mathrm{d}\boldsymbol{y}. \tag{4.4}$$

When $r > r_c$ the induced potential is equal to $\frac{1}{4\pi||\boldsymbol{r}||}$. Depending on the charge distribution, this potential ϕ can be evaluated analytically.

By introducing the 3D Green's function in equation (4.1), the electrostatic potential can be split into a short-range and a long-range contribution [10]

$$\Phi(\boldsymbol{x}) = -\frac{1}{4\pi\epsilon_0}\sum_{\boldsymbol{n}\in\mathbb{Z}^3}\sum_{\substack{i=1\\ \boldsymbol{x}_i^{\boldsymbol{0}}\neq\boldsymbol{x}}}^{N} q_i \times \int_{\mathbb{R}^3}\left(\frac{\delta(\boldsymbol{y}-\boldsymbol{x}_i^{\boldsymbol{n}}) - \varphi_i^{\boldsymbol{n}}(\boldsymbol{y})}{||\boldsymbol{x}-\boldsymbol{y}||} + \frac{\varphi_i^{\boldsymbol{n}}(\boldsymbol{y})}{||\boldsymbol{x}-\boldsymbol{y}||}\right)\mathrm{d}\boldsymbol{y}. \tag{4.5}$$

The short-range term is given by

$$\Phi^{nf}(\boldsymbol{x}) := -\frac{1}{4\pi\epsilon_0}\sum_{\boldsymbol{n}\in\mathbb{Z}^3}\sum_{\substack{i=1\\ \boldsymbol{x}_i^{\boldsymbol{0}}\neq\boldsymbol{x}}}^{N} q_i \times \left(\frac{1}{||\boldsymbol{x}-\boldsymbol{x}_i^{\boldsymbol{n}}||} - \int_{\mathbb{R}^3}\frac{\varphi_i^{\boldsymbol{n}}(\boldsymbol{y})}{||\boldsymbol{x}-\boldsymbol{y}||}\mathrm{d}\boldsymbol{y}\right). \tag{4.6}$$

$$= \sum_{\boldsymbol{n}\in\mathbb{Z}^3}\sum_{\substack{i=1\\ \boldsymbol{x}_i^{\boldsymbol{0}}\neq\boldsymbol{x}}}^{N} C(\boldsymbol{x}_i^{\boldsymbol{n}},\boldsymbol{x}) \tag{4.7}$$

where

$$\begin{aligned} C(\boldsymbol{x},\boldsymbol{x}_i^{\boldsymbol{n}}) &= -\frac{q_i}{\epsilon_0}\times\left(\frac{1}{4\pi||\boldsymbol{x}-\boldsymbol{x}_i^{\boldsymbol{n}}||} - \int_{\mathbb{R}^3}\frac{\varphi_i^{\boldsymbol{n}}(\boldsymbol{y})}{4\pi||\boldsymbol{x}-\boldsymbol{y}||}\mathrm{d}\boldsymbol{y}\right) \\ &= -\frac{q_i}{\epsilon_0}\times\left(\frac{1}{4\pi||\boldsymbol{x}-\boldsymbol{x}_i^{\boldsymbol{n}}||} - \phi(||\boldsymbol{x}-\boldsymbol{x}_i^{n}||)\right) \end{aligned} \tag{4.8}$$

measures the error introduced while replacing a point charge located at $\boldsymbol{x}_i$ by a smooth density. As in P3M, the main advantage of introducing locally smeared charge distributions is the fact that they behave like Dirac densities beyond the cutoff distance: when $||\boldsymbol{x}-\boldsymbol{x}_i^{\boldsymbol{n}}|| > r_{max}$, $C(\boldsymbol{x},\boldsymbol{x}_i^{\boldsymbol{n}}) = 0$. As a result, the short-range contribution can be evaluated by taking into account only the interactions up to a given cutoff distance r_c [10, 206]. Let us define $\mathcal{N}(\boldsymbol{x}) := \{\boldsymbol{x}_i^{\boldsymbol{n}}, \boldsymbol{n}\in\mathbb{Z}^3 / ||\boldsymbol{x}-\boldsymbol{x}_i^{\boldsymbol{n}}|| < r_c; \boldsymbol{x}_i^{\boldsymbol{n}}\neq\boldsymbol{x}\}$ as the set of all particles (including periodic images) in the neighborhood of $\boldsymbol{x}$. Therefore, using equation (4.4), one can express this short-range contribution as follows:

$$\begin{aligned} \Phi^{nf}(\boldsymbol{x}) &= -\frac{1}{\epsilon_0}\sum_{\boldsymbol{x}_i\in\mathcal{N}(\boldsymbol{x})} q_i \times\left(\frac{1}{4\pi||\boldsymbol{x}-\boldsymbol{x}_i||} - \phi(||\boldsymbol{x}-\boldsymbol{x}_i||)\right) \\ &= \sum_{\boldsymbol{x}_i\in\mathcal{N}(\boldsymbol{x})} C(\boldsymbol{x},\boldsymbol{x}_i) \end{aligned} \tag{4.9}$$

The computation of the short-range contribution can be handled efficiently with a neighbor list, which results in an optimal $\mathcal{O}(N)$ scaling.

Algorithm 3 Near-field correction $\Phi^{nf}(\boldsymbol{x}_i)$

1: **for** particle $\boldsymbol{j}/\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i)$ **do**
2: $c \leftarrow -\dfrac{q_j}{4\pi\epsilon_0||\boldsymbol{x}_i-\boldsymbol{x}_j||} + \dfrac{q_j}{\epsilon_0}\phi(||\boldsymbol{x}_i-\boldsymbol{x}_j||)$ ▷ Compute the correction $C(i,j)$
3: $\Phi^{nf}(\boldsymbol{x}_i) \leftarrow \Phi^{nf}(\boldsymbol{x}_i) + c$
4: **end for**

4.2.3 The smooth component

The smooth long-range part in (4.5) corresponds to the solution of the modified version of equation (4.2) where Dirac impulses were replaced by a smooth distribution.

$$\Delta\Phi^{sm}(\boldsymbol{x}) = -\frac{1}{\epsilon_0}\sum_{i=1}^{N} q_i \varphi_i^{\boldsymbol{n}}(\boldsymbol{x}) \tag{4.10}$$

The new formulation can be solved with any grid-based Poisson solver. This equation is discretized on a Cartesian grid Ω^h of spacing h. A fourth-order compact discretization of the equation (4.10) derived by Stephenson [8] leads to a linear system of equations, which can be solved using a multigrid algorithm. The PDE (4.10) can be solved in diverse ways. Nevertheless, due to its good scaling and its linear computational complexity, the multigrid approach is the method of choice for this work. This multi-resolution method speeds up the convergence of a basic iterative method (Gauss-Seidel, Jacobi, etc.) which efficiently reduces the high-frequency error. The multigrid method defines a hierarchy of coarser and coarser grids on which the low-frequency error at a given level is seen as a high-frequency one at a coarser level (see *e.g.* Trottenberg et al. [209] or Brandt et al. [30] or section 2.5).

4.2.4 The self-component

Sum (4.1) for the potential does not contain the self interaction $\boldsymbol{x} = \boldsymbol{x}_i$. In fact, the electrostatic potential due to a point charge diverges at the source's location. Conversely, a smooth charge distribution does not lead to a divergence of the potential. Moreover, the formulation (4.10) implies that Φ^{sm} sums the contributions of all charges including eventually $\boldsymbol{x} = \boldsymbol{x}_i$. Therefore, at each particle's location one must subtract a term corresponding to a self-interaction. This term, here denoted by Φ^{self}, is proportional to $\phi(\mathbf{0})$:

$$\Phi^{self}(\boldsymbol{x}_i) = \frac{q_i}{\epsilon_0}\phi(\mathbf{0}). \tag{4.11}$$

Equation (4.4) yields

$$\phi(\mathbf{0}) = \int_{\mathbb{R}^3} \frac{\rho_{r_c}(||\boldsymbol{y}||)}{4\pi||\boldsymbol{y}||}\mathrm{d}\boldsymbol{y}$$

Here, the integrand is a radial function. Therefore, a change of variables gives

$$\begin{aligned} \phi(\mathbf{0}) &= \int_{\mathbb{R}^+} r\rho_{r_c}(r)\mathrm{d}r \\ &= \int_0^{r_c} r\rho_{r_c}(r)\mathrm{d}r. \end{aligned} \tag{4.12}$$

Finally, the potential on a given particle i can be expressed as

$$\Phi(\boldsymbol{x}_i) = \Phi^{nf}(\boldsymbol{x}_i) + \Phi^{sm}(\boldsymbol{x}_i) - \Phi^{self}(\boldsymbol{x}_i). \tag{4.13}$$

4.2.5 Interpolation at particles positions

For a particle i, one can retrieve the smooth term $\Phi^{sm}(\boldsymbol{x}_i)$ by interpolating Φ_g^{sm} at the particle position $\boldsymbol{x}_i$, where Φ_g^{sm} is the solution of (4.10) on a grid

$$\Phi^{sm}(\boldsymbol{x}_i) \approx \tilde{\Phi}^{sm}(\boldsymbol{x}_i) = \sum_{\boldsymbol{m} \in \mathscr{I}(\boldsymbol{x}_i)} \omega_{\boldsymbol{m}}^{(i)} \Phi_g^{sm}(\boldsymbol{m}) \tag{4.14}$$

where $\mathscr{I}(x_i) \subset \Omega^h$ contains neighboring grid points of particle $\boldsymbol{x}_i$ and $\omega_{\boldsymbol{m}}^{(i)}$ are the associated weights. From the P closest grid nodes of particle i in each direction, a sub-mesh of P^3 neighboring points of i can be obtained. A similar stratagem was employed for P3M , where electrostatic forces were approximated with weights computed with splines[54]. In this work, $\Phi^{sm}(\boldsymbol{x}_i)$ was determined with a three-dimensional polynomial interpolation [72, 188]. The associated weights $\omega_{\boldsymbol{m}}^{(i)}$ depend on the atomic relative positions with respect of the grid. In addition, they are obtained by computing the tensor product of interpolation coefficients derived from the interpolation in each dimension.

The meshed continuum method can be summarized by the following algorithm:

Algorithm 4 Compute electrostatic potential $\Phi(\boldsymbol{x}_i)$

$\rho^{sm} \leftarrow 0$ on the grid Ω^h
for each particle $\boldsymbol{i}$ **do**
    compute the near-field correction $\Phi^{nf}(\boldsymbol{x}_i)$ ▷ *see* algorithm 3
    $\Phi^{self}(\boldsymbol{x}_i) \leftarrow \frac{q_p i}{\epsilon_0} \phi(\mathbf{0})$ ▷ Self-correction
    Add $\boldsymbol{i}$'s contribution to smooth distribution ρ^{sm} on the grid Ω^h
end for
Solve $\Delta\Phi^{sm} = \rho^{sm}$ via a multigrid method
for each particle $\boldsymbol{i}$ **do**
    $\Phi^{sm}(\boldsymbol{x}_i) \leftarrow 0$
    for grid point $\boldsymbol{m} \in \mathscr{I}(\boldsymbol{x}_i)$ **do**
        $\Phi^{sm}(\boldsymbol{x}_i) \leftarrow \Phi^{sm}(\boldsymbol{x}_i) + \omega_{\boldsymbol{m}}^{(i)} \Phi^{sm}(\boldsymbol{m})$ ▷ Interpolate Φ^{sm}
    end for
    $\Phi(\boldsymbol{x}_i) \leftarrow \Phi^{nf}(\boldsymbol{x}_i) + \Phi^{sm}(\boldsymbol{x}_i) - \Phi^{self}(\boldsymbol{x}_i)$ ▷ Retrieve the electrostatic potential
end for

4.3 Incremental Meshed Continuum Method

Despite its linear scaling, the Meshed Continuum Method is slower than $\mathcal{O}(N\log(N))$ mesh solvers in Fourier space. Nevertheless, this $\mathcal{O}(N)$ complexity suggests that this method should be suitable for an incremental update. The evaluation of the right-hand side of the equation (4.2) and the computation of the near-field correction are the most expensive components of the algorithm. In this section, we propose an incremental version of the MCM method which takes advantage of ARMD.

Given two successive timesteps ($t = t_0$) and ($t = t_1$), particles have two kinds of dynamics in ARMD. Whilst the so-called restrained charges are frozen between two successive timesteps, the active ones are allowed to move freely (*see* figure 4.1). Our goal is to take advantage of the fact that some charges are restrained from t_0 to t_1 in order to incrementally update the potential Φ at t_1 knowing its value at t_0, thus speeding up the calculation of long-range forces.

Let us assume that at the timestep $t = t_0$, R particles are restrained (frozen). Other particles are said to be active and able to move freely. Knowing the potential at $t = t_0$, we want to speed up the computation of the potential at $t = t_1$, knowing that the R restrained particles keep the same positions at $t = t_1$. The proposed algorithm scales linearly with the number of active particles $A = N - R$.

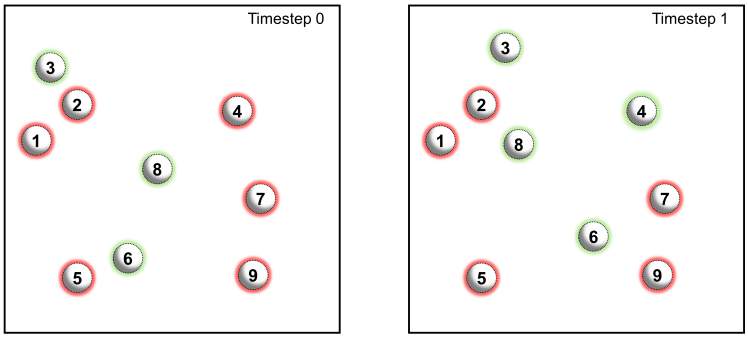


Figure 4.1: Two consecutive timesteps in ARMD. Left corresponds to ($t = t_0$) and right to ($t = t_1$). Red particles are restrained and cannot move. Green particles are active and are allowed to move freely.

4.3.1 Incremental computation of the right-hand side

The calculation of the right-hand side (r.h.s.) of the equation (4.10) can easily be handled with a $\mathcal{O}(N)$ algorithm, since the number of grid points affected by each particle is smaller than the total number of grid points. The incremental calculation of the right-hand side is thus straightforward by splitting the potential into restrained and active contributions:

$$\rho^{sm} = \rho^{(Active)} + \rho^{(Restrained)} \tag{4.15}$$

At $t = t_0$ the term $\rho^{(Restrained)}$, which corresponds to the r.h.s. associated to restrained particles, is stored, and does not need to be updated at $t = t_1$. Only $\rho^{(Active)}$ needs to be computed at each timestep. Assuming the overhead resulting from (4.15) is small, the calculation has an $\mathcal{O}(A)$ complexity. In practice, two separate grids ($\mathcal{G}_{all}$ and $\mathcal{G}_{restrained}$) have to be used. $\mathcal{G}_{restrained}$ holds the contribution $\rho^{(Restrained)}$ obtained by running over all restrained particles in the system and sum successively charge contributions on the grid according to their relative location on the mesh. $\mathcal{G}_{all}$ corresponds to the contribution of all particles ρ^{sm}. At the initial timestep, $\mathcal{G}_{restrained}$ is computed and saved in memory. For each timestep, one can copy values from $\mathcal{G}_{restrained}$ into $\mathcal{G}_{all}$. Then, the contribution of all active particles is added to $\mathcal{G}_{all}$ in order to obtain ρ^{sm}. Consequently, this strategy may significantly reduce the CPU cost of the evaluation of ρ^{sm} by doubling the required memory.

In an adaptively restrained simulation, the distribution of active and restrained particles is not constant. At each time step, some restrained particles may become active, and vice versa (*e.g.* Particle 4 in Figure 4.1). At the next timestep $t = t_2$, the restrained component thus needs to be updated. This leads to a $\mathcal{O}(S)$ task, where S is the number of switching particles. Typically, $S << A < N$, and the extra cost can be neglected.

4.3.2 Incremental near-field correction

Singh and Redon [190] proposed an incremental algorithm to tackle the computation of short-range pairwise potential in ARMD . The near-field correction presented in the algorithm 3 can be seen as a short-range pairwise potential. In fact, from (4.8), the total of near-field corrections needed to be applied to a given particle i can be obtained by the sum $C(i,:) = \sum_j C(i,j)$, where i and j are neighbors.

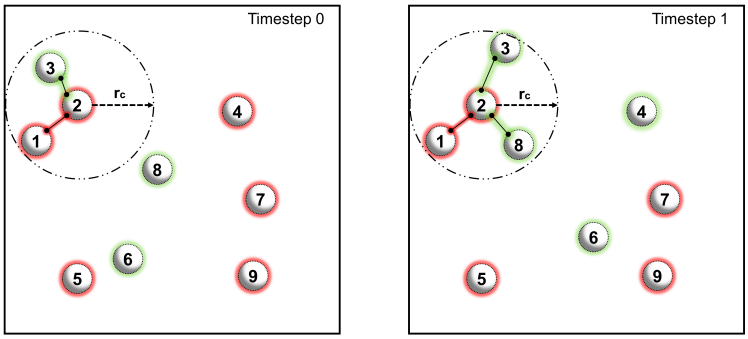


Figure 4.2: Near correction for particle (2) at ($t = t_0$) (left) and ($t = t_1$) (right). **Restrained-Restrained** corrections (red links) are unmodified between two successive time steps. **Active-Restrained** corrections (green links) have to be updated.

In the considered system, two kinds of corrections have to be taken into account (Figure 4.2):

- The correction where both particles are *restrained*: **Restrained-Restrained** corrections
- The correction where at least one particle is *active*: **Active-Restrained** or **Active-Active** corrections

Restrained-Restrained corrections can be stored between two consecutive timesteps, since their value does not change. Conversely, other corrections need to be recomputed. Therefore, one can split $C(i,:)$ as a sum of **Restrained-Restrained** corrections on particle i ($C(i,\mathscr{R})$) and **Active-Restrained** or **Active-Active** corrections on particle i ($C(i,\mathscr{A})$):

$$
\begin{aligned}
C(i,:) &= C(i,\mathscr{R}) + C(i,\mathscr{A}) \\
&= \sum_{j\in\mathscr{N}(i)} C(i,j)\mathbb{1}^{\mathscr{R}}(i)\mathbb{1}^{\mathscr{R}}(j) + \sum_{j} C(i,j)(1-\mathbb{1}^{\mathscr{R}}(i)\mathbb{1}^{\mathscr{R}}(j)).
\end{aligned} \tag{4.16}
$$

Here $\mathbb{1}^{\mathscr{R}}$ corresponds to the characteristic function of the set of restrained particles.

While $C(i,\mathscr{R})$ remains unchanged between $t = t_0$ and $t = t_1$, $C(i,\mathscr{A})$ needs to be computed. When the particle i is restrained, $C(i,\mathscr{R})$ corresponds to the sum of corrections that involve only restrained neighbors of i and $C(i,\mathscr{A})$ sums the contributions from its active neighbors. Obviously, when i is active, $C(i,\mathscr{R})$ is nil and $C(i,\mathscr{A})$ is obtained by summing the contributions from all neighbors of i.

The superfluous computation of interactions between restrained particles can be avoided by storing for each particle the **Restrained-Restrained** component of the correction (*see* algorithm 5). For a restrained particle, the missing contribution can be computed at each timestep by looping only on its *active* neighbors. Furthermore, the incremental near-field correction can be done more efficiently by modifying the neighbor list of each particle such that it carries only the information on the needed pairwise corrections [191].

Algorithm 5 Compute the **Restrained-Restrained** component $\Phi^{(nf,R)}(\boldsymbol{x}_i)$

if $\boldsymbol{i}$ is *restrained* **then**
  for **restrained** particle $\boldsymbol{j}/\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i)$ **do**
    $\Phi^{(nf,R)}(\boldsymbol{x}_i) \leftarrow \Phi^{(nf,R)}(\boldsymbol{x}_i) - \dfrac{q_j}{4\pi\epsilon_0||\boldsymbol{x}_i - \boldsymbol{x}_j||} + \dfrac{q_j}{\epsilon_0}\phi(||\boldsymbol{x}_i - \boldsymbol{x}_j||)$
  end for
else
  $\Phi^{(nf,R)}(\boldsymbol{x}_i) \leftarrow 0$
end if

As mentioned above, usually there is a small number of particles which switch state between consecutive timesteps. A special treatment needs to be done for these particles. At each timestep, the correction between switched particles and restrained particles must be removed from or added to the **Restrained-Restrained** component depending on the nature of the switch (« *Restrained* to *Active* » or « *Active* to *Restrained* »). Therefore the incremental near correction is an $\mathcal{O}(A+S)$ task.

Algorithm 6 Update for switched particles

for each *switched* particle $\boldsymbol{s}$ **do**
  if $\boldsymbol{s}$ is *restrained* –> *active* **then**
    Remove contribution to $\rho^{(sm,R)}$
    $\Phi^{(sr,R)}(\boldsymbol{x}_s) = 0$
  else
    Add contribution to $\rho^{(sm,R)}$
    Compute $\Phi^{(nf,R)}(\boldsymbol{x}_s)$ ▷ *see* Algorithm 5
  end if
end for

4.3.3 Interpolation coefficients

A final word is on the computation of the interpolation coefficients $(\omega_{\boldsymbol{m}}^{(i)})_{\boldsymbol{m}\in\mathscr{I}(\boldsymbol{x}_i)}$. The computation of these weights can be done on-the-fly as in the original Meshed Continuum Method. However, we chose to precompute them at the first timestep. In practice, since these weights can be obtained through tensor product, only 1-dimensional coefficients are stored. The memory requirement of this approach is $3P$ coefficients per particle (instead of P^3). A slight overhead is therefore generated, but this approach might be useful on large systems. In addition, saving interpolation coefficients is a suitable option for Adaptively Restrained Molecular Dynamics. In fact, these weights can be reused as long as the corresponding particle does not move. As a consequence, only coefficients associated to *active* particles require a frequent update.

Withal, the incremental MCM is summarized with the algorithm 7.

Algorithm 7 Incremental electrostatic potential $\Phi(\boldsymbol{x}_i)$

Require: $(\omega_{\boldsymbol{m}}^{(i)})_{\boldsymbol{m}\in\mathscr{I}(\boldsymbol{x}_i)}$, $\Phi^{(nf,R)}(\boldsymbol{x}_i)$, $\rho^{(sm,R)}$, $\Phi^{self}(\boldsymbol{x}_i)$.

1: **for each** *restrained* particle $\boldsymbol{r}$ **do**
2: $\quad\Phi^{nf}(\boldsymbol{x}_r) = \Phi^{(nf,R)}(\boldsymbol{x}_r)$
3: $\quad$Update near-field correction for only active neighbors $\Phi^{nf}(\boldsymbol{x}_r)$
4: **end for**
5: $\rho^{sm} \leftarrow \rho^{(sm,R)}$ on the grid Ω^h
6: **for each** *active* particle $\boldsymbol{a}$ **do**
7: $\quad$Add contribution to smooth distribution ρ^{sm} on the grid Ω^h
8: $\quad$Update interpolation weights $(\omega_{\boldsymbol{m}}^{(a)})_{\boldsymbol{m}\in\mathscr{I}(\boldsymbol{x}_a)}$
9: $\quad$Update near-field correction $\Phi^{nf}(\boldsymbol{x}_a)$ ▷ *see* Algorithm 3
10: $\quad\Phi^{self}(\boldsymbol{x}_a) \leftarrow q_a\phi(0)$ ▷ Update self-correction
11: **end for**
12: Solve $\Delta\Phi^{sm} = \rho^{sm}$ via multigrid method
13: Add all contributions to $\Phi(\boldsymbol{x}_i)$ ▷ Analogous to lines [8-14] of Algorithm 4
14: Update *active* and *restrained* lists
15: Update for switched particles ▷ *see* Algorithm 6

4.3.4 Parallelization

We implemented the Incremental Meshed Continuum Method (IMCM) in the LAMMPS [167, 169], a well-established simulation package. For completeness, this section presents a brief overview of the parallelization capabilities of the presented methods.

The meshed continuum method can be parallelized efficiently with the help of a standard domain-decomposition scheme where the physical domain, partitioned as sub-

domains, is distributed onto the available processors. LAMMPS exploits the Message Passing Interface (MPI) standard and the Recursive Coordinate Bisection algorithm to distribute sub-domains (i.e. subsets of particles and grid points) over MPI processes. Usually, all computations required by a geometric multigrid method can be described with the help of stencil operators, which often have a compact support [96]. Thus, each process can apply these stencils on grid points located inside its sub-domain. However, for a given process, the computations on grid points that are situated close to the sub-domain boundary may require grid points that are held by nearby sub-domains. Therefore, to ensure that the necessary data is available during computations, each sub-domain is enlarged by a surrounding *ghost* area.

To evaluate the right-hand side of Equation (4.10), each process computes the needed contribution on the grid points it owns (including ghost ones). Then, a *backward* communication step is applied. In order to fully sum contributions in their domains, all processes communicate the quantities of their ghost grid points and accumulate them into their own *real* grid points. Periodic boundary conditions are taken into account if needed.

Near-field contributions obey the same principles. They share the same computational pattern as short-range potentials in LAMMPS. Each process computes interactions between its particles (both non-ghost and ghost) through neighbor lists. Then, ghost particles contributions are communicated to the corresponding non-ghost particles.

The implemented multigrid solver follows the popular V-Cycle strategy [33, 209]. In order to travel through the hierarchy of grids defined by the multigrid method, restriction and prolongation operations must be applied. These operations can be formulated as stencil operators and can be locally applied on each sub-domains. Then, a global communication is required after each restriction or prolongation operation[43, 206]. In addition, the discretized Poisson problem can be described, on each grid level, by a stencil operation. Therefore, additional communications are required. Although this parallelization scheme can be handle with ease, the multigrid method requires more attention, especially on large clusters. In fact, as the problem size is reduced on coarser grids levels, it may arise that the number of unknowns exceed the number of available process. Various strategies have been proposed in order to tackle this issue [43]. We choose to let the processes without assigned unknowns stay idle during the computation on these coarser levels. It should be noted that this approach introduces load-imbalance.

Finally, each process retrieves forces by deriving the obtained solution at its own grid points. A *forward* communication step is performed in order to fill the ghost grid cells

surrounding each sub-domain.

Parallelization capabilities presented above can be effortlessly extended to IMCM. For instance, incremental computation of the near-field contributions can be done as follows: each MPI process computes corrections acting on its own particles (including ghosts), while avoiding computations between restrained particles. This is similar to what is proposed in [192] to parallelize the calculation of short-range interactions in ARMD.

4.4 Benchmarks

In order to evaluate the method, we designed three test systems containing sodium (Na) and Chlorine (Cl) particles:

- System A (64000 particles) - corresponds to a solid sodium chloride. In a $112.8 \times 112.8 \times 112.8$ $\mathring{A}^3$ periodic box, 32000 NaCl pairs were placed according to the halite or rock-salt crystal structure. We evaluated our implementation (accuracy and CPU time) of the meshed continuum method by running NVE molecular dynamics on system A. Then, the performance of the Incremental Meshed Continuum Method was assessed. To do so, we ran several simulations where, between two consecutive timesteps, a percentage (ranged from 0 to 100%) of randomly selected particles was restrained. Thus, by mimicking the behavior of ARMD, we could easily understand the acceleration of various components of IMCM. This system was also used to evaluate the parallelization of IMCM. Precisely, we replicate the system twice in each direction (512000 atoms).

- System B (23232 particles) - corresponds to a sodium chloride in the aqueous solution at 10.0 molality concentration. 6912 water molecules were combined with 1248 NaCl pairs in a $65.26 \times 65.26 \times 65.26$ $\mathring{A}^3$ periodic box. The system B was maintained at 298K with a Langevin thermostat [211] and thermodynamic properties were computed. During these simulations, water particles dynamics were adaptively controlled via various energy thresholds $(\varepsilon^r, \varepsilon^f)$. Meanwhile Na and Cl particles are always active — $(\varepsilon^r, \varepsilon^f)_{Na;Cl} = (0;0)$. $(\varepsilon^r, \varepsilon^f)$ affects the average number of restrained particles. Therefore, one may carefully choose them. Thus, we evaluated the ability of the IMCM to speed up adaptively restrained simulations of coulombic systems.

- System C (26290 particles) - Here we studied the interaction of NaCl and water ions with a monolayer porous graphene in cubic box with a $L = 70\,\text{Å}$ edge . A graphene layer with a 7 Å -diameter nanopore is placed at the center of the system in the plane ($z = 0$). An external electric field of $1\,\text{V/Å}$ applied along z-direction drives particles through the nanopore. 6 charged particles are placed at the edge of the pore. Particles in the graphene sheet, except those forming the pore, are charge free (Fig. 4.3). In order to realize the neutrality, 4 extra counter-ions were added in the system. Furthermore, the system also contains 8000 water molecules, 250 sodium chloride pairs and 1780 carbon atoms.

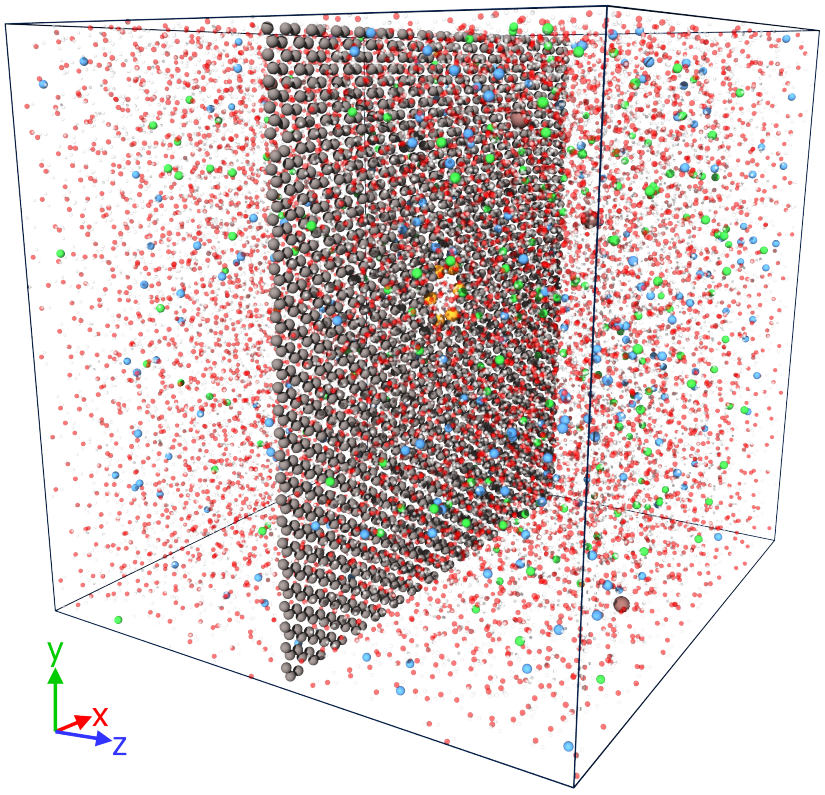


Figure 4.3: Sketch of the nanopore system. Gray color stands for carbon particles of the graphene sheet. Red and white stand for oxygen and hydrogen respectively. Sodium atoms are represented in cyan color while chlorine particles are green. Particles are driven through the pore by an external field applied along z-direction. The flux of sodium atoms is restricted by positively charged particles (orange) located at the edge of the nanopore. Counterions (brown) ensure the neutrality of the system.

During the simulation, the carbon atoms of the graphene layer were frozen without thermal vibration. It was shown that this has a minor impact on the overall

dynamics [95, 202]. Similarly to system B, sodium chloride ions are always active whilst water molecules follow adaptively restrained molecular dynamics. Transport properties of this system were analyzed for different combinations of energy thresholds ($\varepsilon^r, \varepsilon^f$). We expected to observe a so-called concentration polarization layer (CPL) in the vicinity of the graphene layer [95] as well as the ion selectivity of the pore [193, 227].

4.4.1 Interaction Potentials

Since previous quoted systems were related to sodium chloride, we used the NaCl/ϵ force-field proposed by Fuentes-Azcatl and Barbosa [69] to describe intermolecular interactions . The advantages of this non-polarizable force-field are its simplicity and its ability to reproduce experimental results. The employed force-field is based on a set of radial particle-particle pair potentials involving Lennard-Jones (LJ) and electrostatic contributions. They used screening factors in the electrostatic interactions to account for the effect of polarization and they showed that in conjunction with the TIP4P/ϵ or the SPC/ϵ water force-fields, their model had a good agreement with experimental data at 298.15 K [69].

Armed with this conclusion, we described intermolecular interactions between particles included Lennard-Jones and electrostatic potentials with scaled charges, as shown in equation (4.17).

$$U_{ij} = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}\lambda_i\lambda_j \tag{4.17}$$

where q_i represents the electric charge of particle i, r_{ij} is the distance between particle i and particle j , σ_{ij} represents the LJ separation distance and ϵ_{ij} is the depth of the LJ potential well. λ_i is the scaling factor introduced to account for the effect of polarization. The parameters used in this research are summarized in table 4.1.

Atom	q_i (e)	λ_i	σ_{ii} (Å)	ϵ_{ii} (kcal/mol)
Na	1.000	0.885	2.520	0.003
Cl	-1.000	0.885	3.850	0.382
H	0.445	1.000	0.000	0.000
O	-0.890	1.000	3.188	0.169
C	0.000	1.000	3.550	0.074
$^*\mathbf{C}^{(a)}$	1.5	1.000	3.550	0.074
$\mathbf{X}^{(b)}$	-2.25	1.000	3.550	0.074

(a) *C corresponds to charged particle forming the nanopore; (b) X are counter ions.

Table 4.1: Values of potential parameters: C corresponds to charged particle forming the nanopore and X are counter ions.

Assuming that the pure water and the ions potentials are compatible, one may use the Lorentz-Berthelot mixing rules for calculating ϵ_{ij} and σ_{ij} (equation (4.18)).

$$\epsilon_{ij} = \sqrt{\epsilon_{ii}\epsilon_{jj}} \quad ; \quad \sigma_{ij} = \left(\frac{\sigma_{ii} + \sigma_{jj}}{2}\right) \tag{4.18}$$

It is straightforward that the treatment of the coulombic term in (4.17) is similar to the work previously presented (*cf.* algorithm 7). Thus, our method can be employed in order to compute incrementally electrostatic component of (4.17).

Finally, water molecules were treated as rigid body using SPC/ϵ model and harmonic potentials were used for bonds and angles (*see* table 4.2) [69].

Potentials	Equilibrium value	Prefactor
O-H Bond	0.957 Å	450 kcal/mol/Å^2
H-O-H Angle	109.47°	55 kcal/mol/°^{2}

Table 4.2: Parameters of harmonic bond and angle potentials in SPC/ϵ water

4.4.2 Implementation details

The Incremental Meshed Continuum method was implemented in the 1^{st} Feb 2014 LAMMPS [168]. Most simulations were performed on a single core on a Dell Precision M4700 laptop with an Intel® Core ™ i7-3840QM CPU @ 2.80GHz. Parallel Benchmarks were performed using a cluster with 8 nodes equipped with 8/16 CPUs Intel Xeon E5540 and a Gigabit Ethernet network. We run the benchmark both for IMCM and for P3M on one node with different number of processes.

We tested various orders of B-spline distributions for the computation of the right hand side. An alternative to these piecewise functions consists in using high-order polynomial functions on $[0, r_c]$. In most cases, the 5^{th}-order B-spline or the 10^{th}-order polynomial function were the best choices. To speed up these functions calls, we used a look-up table with respect to the squared distance. The use of a look-up table reduces the cost of the computation of the right-hand side by more than a factor 3. We computed the near-field correction section 4.2.2 based on a neighbor-list algorithm implemented within LAMMPS [167]. A similar treatment was done for the Lennard-Jones terms in equation (4.17).

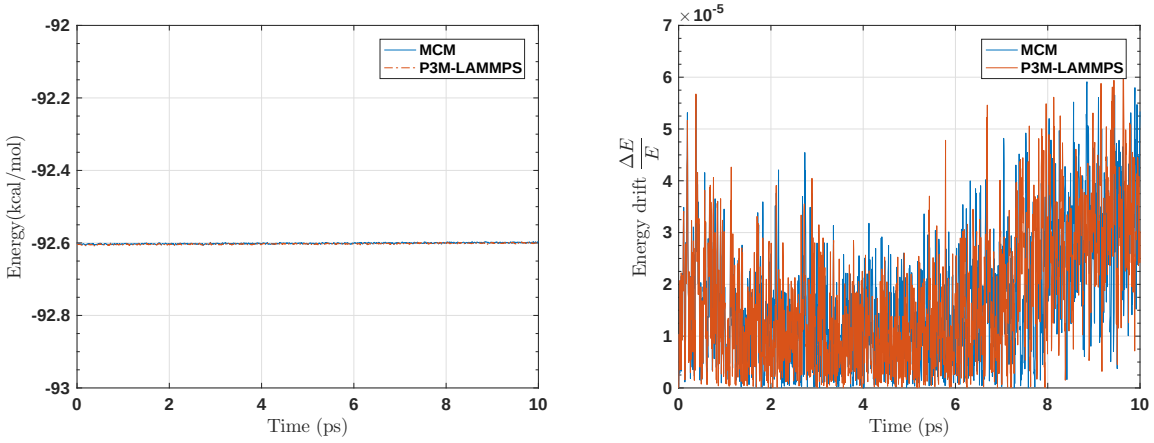
In MD, forces must be computed along with the potential: $F_i = -q_i \nabla \Phi(\boldsymbol{x_i})$. As for particle mesh methods, there is no unique way to achieve this derivation [17]. Analytic differentiation of interpolation weights can be used for this purpose. However, this approach conserves energy but not momentum. One can reduce the momentum drift by removing the mean force. This yields to the non-conservation of energy. A second alternative is to directly derive Φ^{sm} with the use of finite difference operators [67]. Then, the polynomial interpolation scheme is applied to evaluate smooth forces at each particle's position. This approach conserves the momentum while breaking the energy conservation. Fortunately, this drawback can be reduced by applying a mass-weighted correction [195].

In order to validate our implementation, we used P3M to compute reference forces with a 10^{-10} relative error. Thereafter, we evaluated various cutoff radii r_c, grid sizes, interpolation orders, and charge densities (Figure 4.5) to find the optimal configuration for MCM. Finally, we compared the associated CPU processing time to that from P3M solver with tantamount accuracy.

4.5 Results

4.5.1 Solid Sodium Chloride Crystal

MCM was tested with various charge densities (B-splines or polynomials) on the system A using standard molecular dynamics. For this pure NaCl system of 64000 particles, the choice of the 10^{th}-order polynomial as charge density seems to be the best option for MCM (Figure 4.5(1)). Several grid sizes were also evaluated. This leads us to the conclusion that for the system A, the 64^3 grid is the fastest choice for low-accuracy simulations while a 128^3 grid is efficient for high-quality ones. Roughly, the use of a lower grid size will require a larger cutoff radius. Therefore, the processing time of the method will be dominated by the computation of the near-field correction. A bigger grid size allows the use of a smaller cutoff. The cost of the near-field correction is indeed reduced but the overall cost is governed by the calculation of the right-hand side (Figure 4.6) which can be pricey.



(1) Energy per particle.

(2) Energy drift.

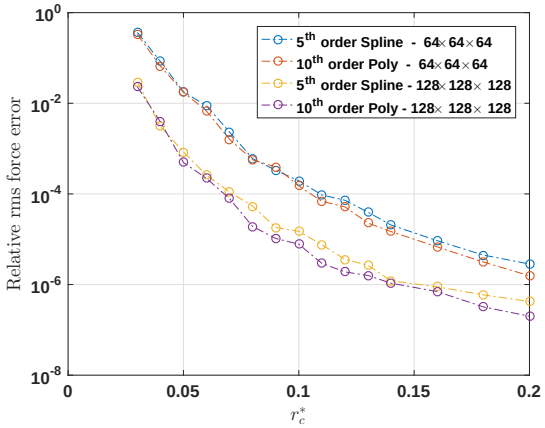
Figure 4.4: The evolution of the energy E per particle over 10ps for a NVE simulation of 64000 particles with time-step increment Δt = 2fs. MCM is compared to P3M at similar accuracy ($\sim 10^{-5}$). The energy drift $|\Delta E/E|$ of both methods is also plotted (right).

NVE simulations were conducted with the meshed continuum method (MCM) in order to confirm the quality of our implementation. For the same order of accuracy, we had similar energy conservation for both P3M and MCM (figure 4.4). However, for a given accuracy MCM is 2 to 4 times slower than P3M (Figure 4.5(2)). These results are slightly better in comparison to the previous state-of-the-art benchmarks [10]. For the

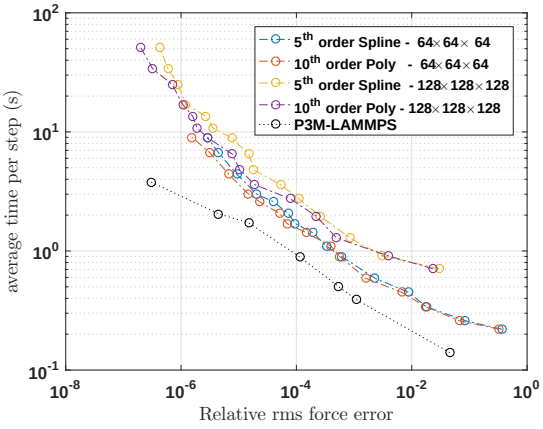
purpose of a fair comparison, P3M was tuned in order to achieve the best performance for the given error criterion.

The incremental version of the meshed continuum method was applied to several configurations of the system A, where some particles are restrained between two consecutive timesteps. Various percentages ranging from 0 to 100% of active particles were tested in order to validate the method. Here, particles, under restraints, were constraints at their location without ARMD. This allows us to explore, with ease, diverse distribution of active particles.

As expected, the IMCM scales with the number of active particles. Furthermore, when a sufficient number of particles are restrained, the IMCM algorithm is able to outperform the standard P3M algorithm (precisely, when fewer than 30% of particles are active in this benchmark (Figure 4.6). For a configuration where less than a tenth of particles are active, one can expect at least a ×2 speedup relatively to the Particle Particle Particle Mesh (P3M).

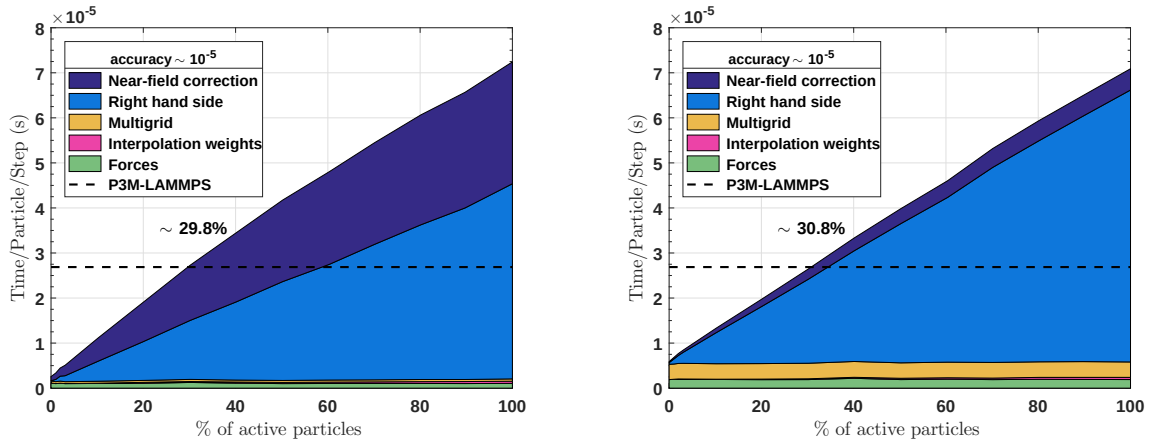


(1) Relative RMS force error versus the cutoff radius.



(2) Required CPU time versus relative RMS force errors.

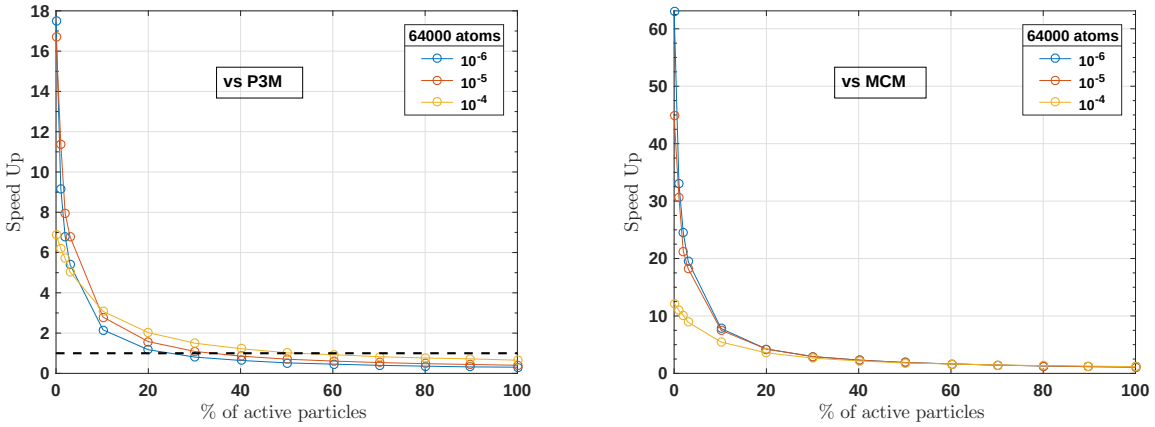
Figure 4.5: Evaluation of our implementation of MCM in LAMMPS on System A (64000 particles). The 10^{th} order polynomial function and the 4^{th} order B-spline were compared on 64^3 and 128^3 Grid. $r_c^* = \frac{r_c}{L}$ is the normalized radius of the charge density.



(1) (2)

Figure 4.6: System A - Runtime of Incremental Meshed Continuum. Left : 64^3 grid and normalized cutoff $r_c = 0.16$. Right: 128^3 grid and normalized cutoff $r_c = 0.09$.

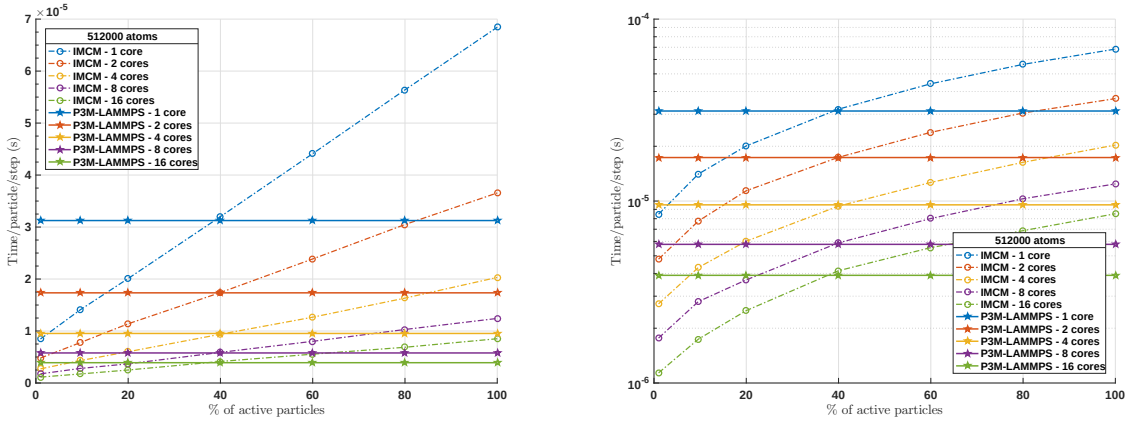
Figure 4.7 confirms that the IMCM is a valuable alternative to P3M methods when an update of forces is needed. When the system is simulated at a relatively good accuracy ($\sim 10^{-6}$), the speedup is more important (*e.g* $\times 5$ with less than 5% actives particles). Moreover, one can also expect a good behavior of the method in Monte Carlo simulations (MC) since the force/energy update via IMCM leads to a $\times 15$ speed-up when 1 or 2 particles are active. This corresponds to a typical scenario of energy computation in a MC trial move.



(1) Speedup of IMCM compared to P3M (2) Speedup of IMCM compared to MCM

Figure 4.7: System A - Speedup of the Incremental Meshed Continuum compared to Particle Particle Particle Mesh (left) and the Meshed Continuum Method (right).

4.5.2 Parallel Performance



(1) Lin-lin scale

(2) Lin-log scale

Figure 4.8: Required wall clock time per particle as a function the percentage of active particles for different number of processes. The wall clock time is represented with Linear scale (left) and logarithm scale (right). Performance of LAMMPS P3M is shown as a reference (dotted lines, pentagram marker) — it does not depend on the percentage of active particles. In all cases electrostatics were computed at similar accuracy ($\sim 10^{-5}$).

We assessed the parallel performance of the IMCM on the replicated version of the system A (512000 atoms). IMCM showed a good behavior on a multi-core environment (Figure 4.8). A 10^{-5} accuracy P3M was used as reference. Linear scale is shown on left In all tests, IMCM becomes faster than P3M when less than about 40% of particles are active. When less than 20% of particles are active, IMCM is 1.5 times faster than P3M on 16 cores. Furthermore, in all configurations, the proposed method is more than twice faster than P3M when the percentage of active particles does not exceed 10%. These results are analogous to the parallel performance of ARMD on short-range potentials[192].

The following sections will be dedicated to several practical applications of ARMD where electrostatic interactions were incrementally updated with IMCM.

4.5.3 Sodium chloride with NaCl/ϵ and SPC/ϵ water

The system of sodium chloride in aqueous solution was studied with ARMD. The system was kept at the temperature $T = 298\,\mathrm{K}$ with a Langevin thermostat. Then, we compared several combinations of energy thresholds ($\varepsilon^r, \varepsilon^f$).

After 10 ns run, the corresponding thermodynamic and dynamic properties of the system were checked against a standard MD simulations at the same order of accuracy. To verify the correctness of our simulations, the hydration of the ions, the structure of the water molecules around ions is computed and checked against results from a standard MD simulations in the same order of accuracy. This hydration can be measured by the four partial radial pair distributions functions (RDFs) g_{Na-H}, g_{Cl-H}, g_{Na-O} and g_{Cl-O}. Although each combination $(\varepsilon^r, \varepsilon^f)$ corresponds to a different average number of active/restrained particles (Section 4.5.3) and a distinct temperature profile (Figure 4.9), the corresponding hydration is similar to results from the reference MD simulation (Figure 4.10). The peak positions, r_{max}, of the pair distribution functions in our model are given by: $r_{max} \simeq 2.98$ Å for Na-H, $r_{max} \simeq 2.19$ Å and $r_{max} \simeq 3.45$ Å for the first and second peaks of Cl-H, $r_{max} \simeq 2.31$ Å for Na-O and $r_{max} \simeq 3.11$ Å for Cl-O.

The computation of the electrostatics forces was sped up with the use of the IMCM. When the average percentage of active particles is around 15%, we achieved ×2 speed up relative to a standard MD run. The performance of the method can be increased by restraining not only water particles but also the sodium chloride ions.

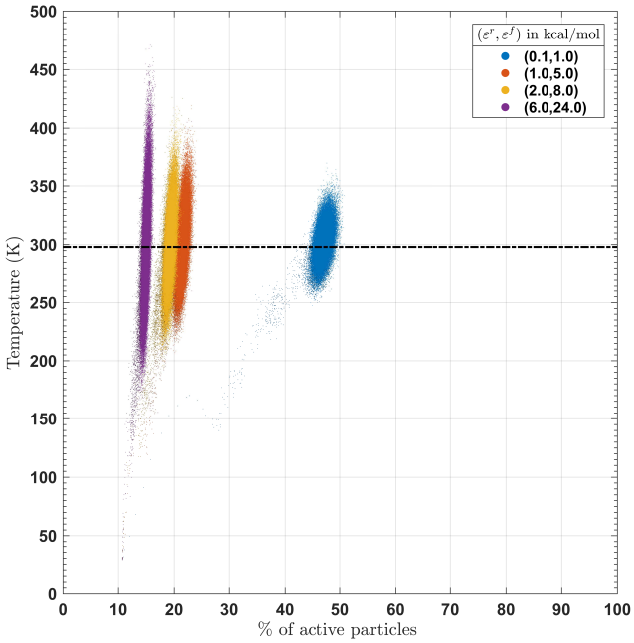
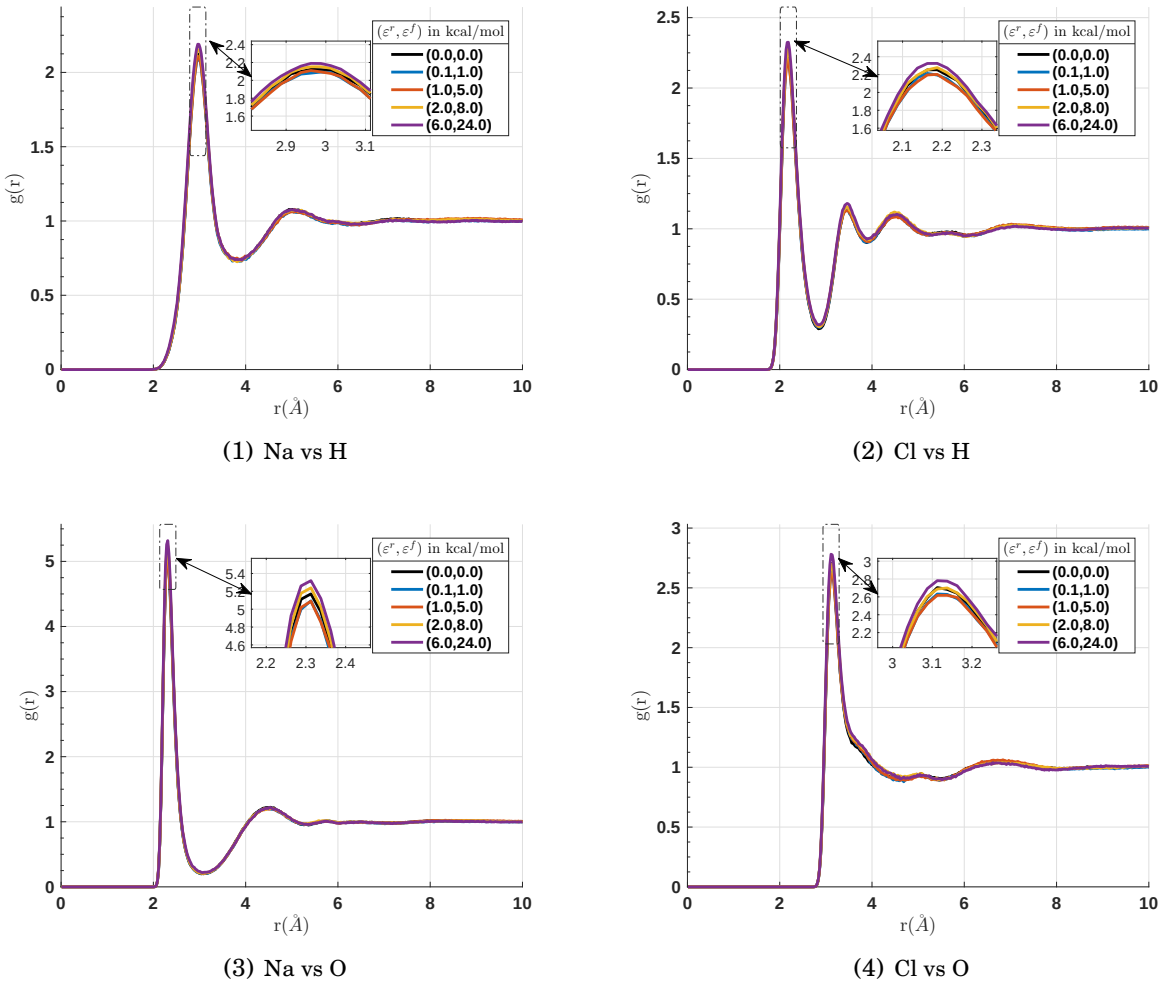


Figure 4.9: Temperature profile for NaCl + water mixture with an ionic concentration of 10.0 molal. Different restraining parameters $(\varepsilon^r, \varepsilon^f)$ were tested on water molecules. Na and Cl are always active. Black line corresponds to the target temperature (298 K).

$(\varepsilon^r, \varepsilon^f)$ (kcal/mol)	$\langle T \rangle$(K)	$\langle n_{act} \rangle$ (%)	Speedup
(0.1,1.0)	298.04	47.02	× 0.90
(1.0,5.0)	298.97	21.56	× 1.54
(2.0,8.0)	297.48	19.37	× 1.63
(6.0,24)	298.73	15.05	× 2.19

Table 4.3: Speed-up for various combinations of energy thresholds $(\varepsilon^r, \varepsilon^f)$. $\langle n_{act} \rangle$ is the average percentage of active particles. $\langle T \rangle$, corresponds to the average temperature.



(1) Na vs H

(2) Cl vs H

(3) Na vs O

(4) Cl vs O

Figure 4.10: Ion-water pair distribution functions using ARMD with the NaCl/ϵ force field at 298 K the rigid water model SPC/ϵ and an ionic concentration of 10.0 molal. Different restraining parameters $(\varepsilon^r, \varepsilon^f)$ were tested on water molecules. Na and Cl are always active. Black line corresponds to a standard molecular dynamics simulation of the system.

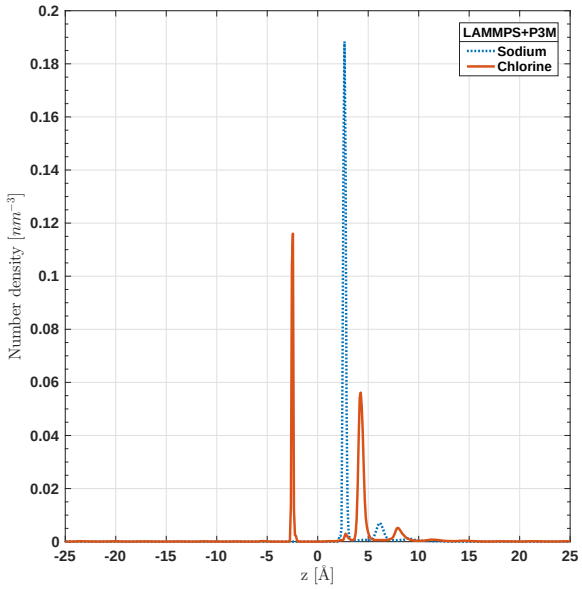
4.5.4 Nanopore System

We investigated the ion transport through a graphene nanopore. As aforementioned, a porous graphene monolayer is placed in a 70^3Å^3 box filled with sodium chloride aqueous solution. Chlorine and sodium ions were driven through the pore with an external electric field perpendicularly oriented to the graphene sheet ($E = 1$ V/Å in minus z-directions). We designed a 5 Å -diameter functionalized pore by putting charges on its edge. Thus, it behaves like an ionic colander of high selectivity [193]. In our case, the nanopore is terminated by positively charged ions, favoring the passage of chlorine. One may produce this type of pore through ion etching [149].

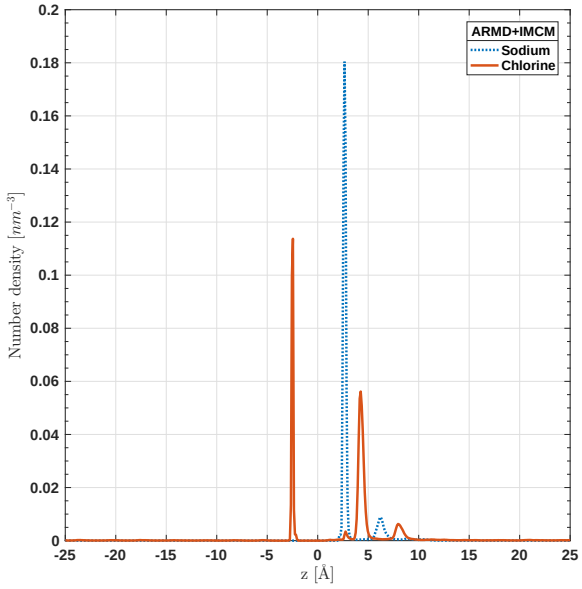
5 ns simulations were conducted with both ARMD and standard MD. The IMCM was employed in ARMD simulations whereas P3M was used in the standard MD runs. In all cases, the timestep was set to be 2 fs. Once again, the system was maintained at 298 K with a Langevin thermostat. The system C was evolved for 2 ns to achieve equilibrium state. Then, statistics were gathered during the last 3 ns. Cylindrical coordinates (r, z) were employed to analyze the behavior of the nanopore system and the origin was set at the center of the simulation box. The initial configuration corresponds to a quick minimization of a system with uniformly distributed sodium chloride ions and without any external field.

When an external electric field is applied, ions are driven directionally and accumulate on the top ($z > 0$) and bottom ($z < 0$) sides of the graphene layer according to the charge's sign. Sodium ions are driven in the (minus z-direction). With periodic boundary conditions, they appeared on the top of the box. Thus, they accumulated in this region since the pore is impermeable to them. In fact, due to charges located at the edge of the pore, sodium ions are repelled and consequently constrained in the top side of the simulation box (Figure 4.11). The combination of electrostatic and Lennard-Jones forces leads to the appearance of this concentration polarization layer (CPL). Conversely, the chlorine ions (*moving from bottom side to topside*) are allowed to pass through the pore. However, chlorine ions formed a concentration polarization layer (CPL) adjacent to the graphene sheet since the pore size is relatively small. The same behavior can be observed for sodium ions. It is interesting to observe that chlorine ions seemed to have three preferential ways to enter the pore. This might be related to the geometry of the designed pore. ARMD in addition to IMCM was able to reproduce this behavior (Figure 4.12), but the average time cost per step is 4 times smaller than standard MD where P3M is used. The ionic distribution of both approaches shows a similar nonuniform ionic distribution with well-defined concentration polarization layers (CPLs) identically located for all ionic

species. ARMD performs quite satisfactorily on this system where approximately 93% of particles were adaptively restrained.



(1) Reference MD



(2) ARMD

Figure 4.11: Number density of chlorine (red dotted line) and sodium (blue dashed line) ions along z-axis using standard MD (Left) and ARMD (Right). Both methods show the ion selectivity of the nanopore which is located at $z = 0$.

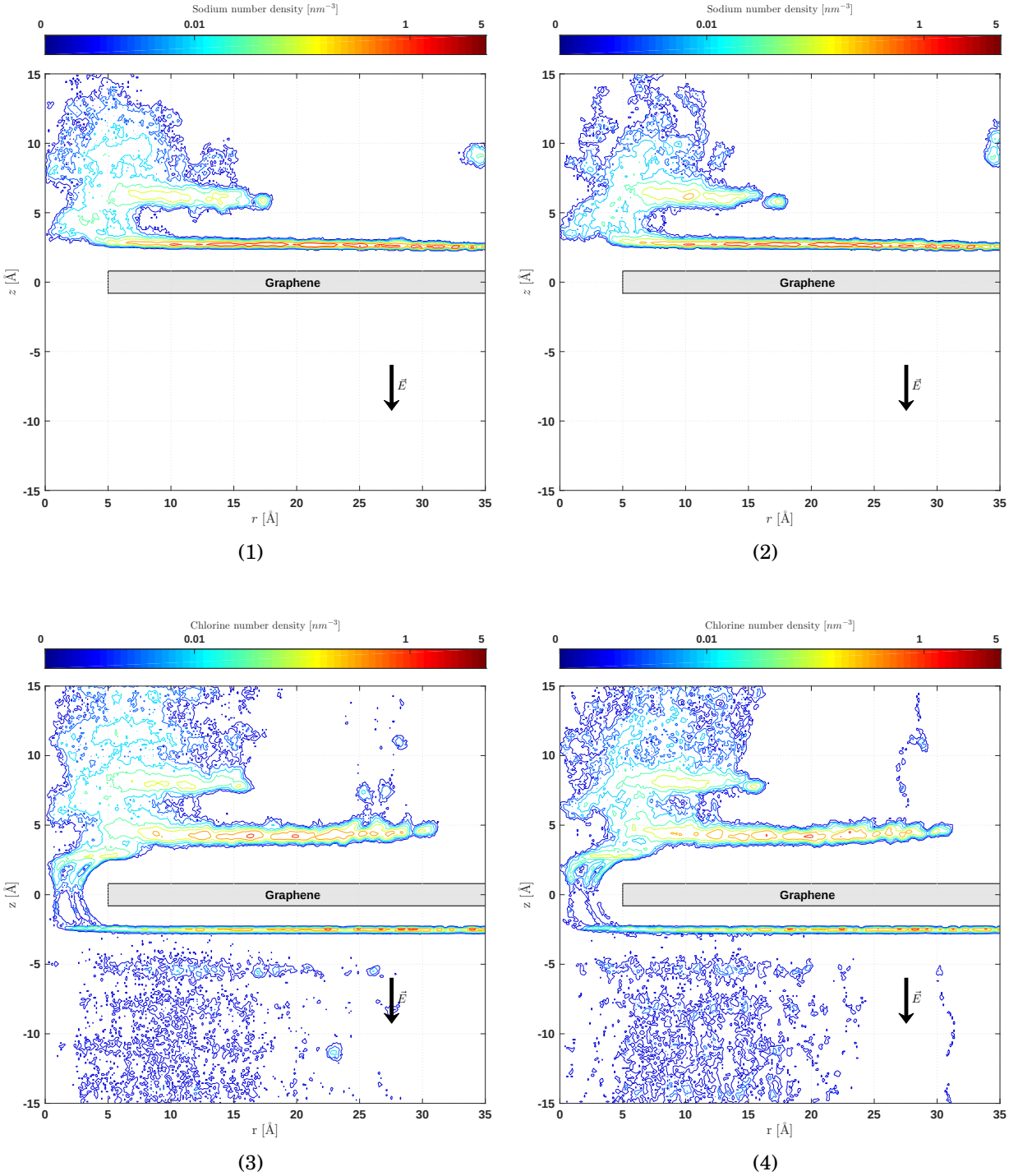


Figure 4.12: Nonuniform distributions of number density of chlorine (Top) and sodium (Bottom) ions driven by an external electric field (black arrow) $E = 1$ V/Å using standard MD (Left) and ARMD (Right). The gray rectangles at $z = 0$ mark the graphene sheet. Both ions form concentration polarization layers (CPLs).

4.6 Conclusion

In this chapter, we proposed a novel approach to the computation of electrostatics forces in adaptively restrained particle simulations. This multigrid-based algorithm takes advantage of the fact that ARMD switches positional degrees of freedom on and off during simulations, while letting momenta evolve [13]. We showed that the pairwise electrostatic potential can be incrementally computed. We achieved significant speedups for adaptively restrained simulations, as the number of active particles decreases. We showed that IMCM exhibits a good behavior on both single and multi-core architectures, and we would like to investigate its performance on massively parallel architectures. We expect that the method will scale well in this context thanks to its multigrid foundation.

We also want to examine several theoretical aspects related to the tuning of the method such as the choice of the smooth distribution, the resolution of the mesh and the cutoff radius.

Moreover the proposed method has shown good performance in terms of precision and speed in comparison to the popular P3M. We believe that the IMCM can be combined with ARMD on a wide range of systems. For instance, we expect that results from section 4.5.1 can be extended to ionic liquids [73]. In addition, combining ARMD and IMCM should allow for a more efficient simulation of channeling effects. Our method can also be used in the simulation of a polymer in a solvent; where a large share of solvent molecules can be restrained. For these reasons, the following chapters will be devoted to certain applications.

Part III

Applications and perspectives

CHAPTER 5

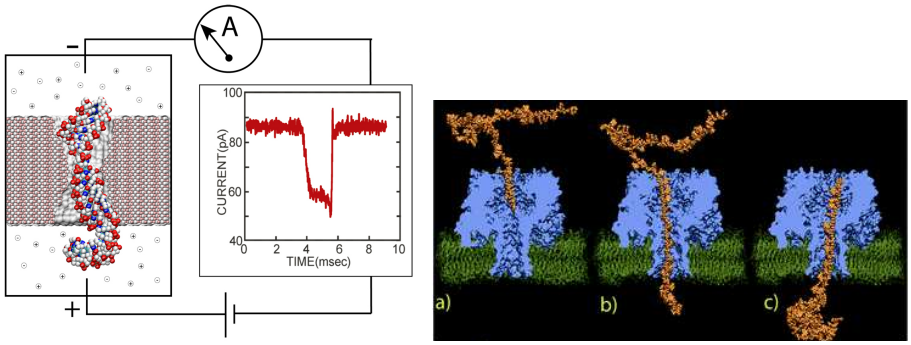
Adaptively Restrained Simulations of a polyelectrolyte translocation

Contents

5.1 Overview

Egged on by a flourishing literature [5, 70, 107, 108, 143, 147, 157, 196], this chapter aims to conduct several ARMD simulations of the translocation of polymers. Since 1994 and the first study of polymers threading through pores by Bezrukov [22], the translocation has been observed in biological phenomena such as the motion of DNA and RNA molecules across nuclear pores, cytoplasmic drug delivery in living cells, DNA packaging into viral capsids, gene swapping and protein transport through membrane channels. For instance, more than half of proteins produced in cells must traverse cellular membranes [124, 152, 157].

In 1996, Kasianowicz et al.[107, 108] have demonstrated that nanopores could be used in order to measure the polynucleotide length and thus characterize DNA. With the help of an electric field, they drove a single-stranded RNA and DNA molecules through a 2.6nm diameter ion channel in a lipid bilayer membrane. The decrease in ionic current induced by the passage of each RNA/DNA molecule is used as a measurement of the polynucleotide length (Figure 5.1).



(1) (2)

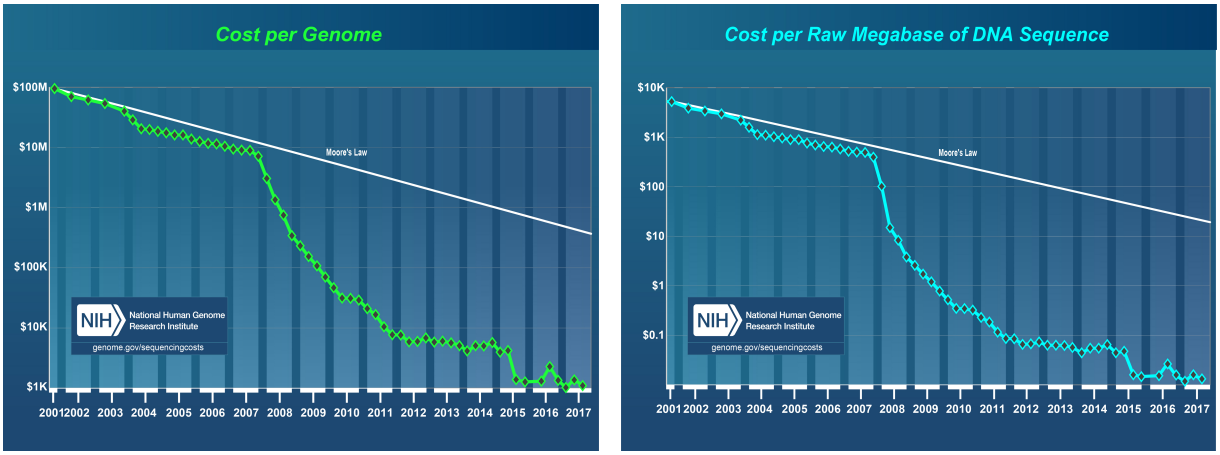
Figure 5.1: MD simulations of DNA translocation performed by Aksimentiev [4]. Experimental setup of a single molecule nanopore experiment (left). Driven by the electrical field between two electrodes, a biomolecule transits the pore in a thin, synthetic membrane inducing a transient decrease of the ionic current. Electric field-driven transport of ssDNA through α-hemolysin (right). More details are available in [4].

Consequently, the translocation of polymers has a great interest in the field of *nanopore sequencing* [5, 31, 70, 217, 222]. Using the nanopore as a biosensor which detects DNA information is a cheap and real-time method of sequencing polynucleotides while preserving their structure (without damaging it). Therefore, nanopore sequencing allows multiple analysis of the same structure. Nowadays, several companies such as

Agilent Laboratories [207], Base4 [125], Illumina [99], Hitachi [89] and Oxford Nanopore [208] are building and commercializing (slim-sized) nanopore sequencers for less than 1000$ (Figure 5.2). Although the commercial success is currently limited, the community is rather enthusiastic as a reduction in costs by a factor 10 is expected in the next decade (Figure 5.3).

(1) SmidgION (2) PromethION

Figure 5.2: Commercial DNA Sequencers: **SmidgION** (left) is a small DNA sequencer from Oxford Nanopore Technologies [208] that can run on a smartphone. **PromethION** (right) is a 1000$ nanopore-based desktop device devoted to DNA sequencing.



(1) (2)

Figure 5.3: Evolution of the cost of DNA sequencing since 2001. Data are available at [104]. Accessed [03/03/2018].

Moreover, the motion of polymers in a confined medium is also technologically important in several industrial processes such as food and medicine production, in oil recovery and separation. For instance, in chemistry, the forced permeation of polymer

molecules and electrophoresis are crucial for the separation and the purification of synthetic (as well as biological) macromolecules. Withal, the polymer translocation becomes a flourishing research topic in nanotechnology.

However, despite all efforts, the physical nature of this process is still not well understood [143, 156, 157]. In particular, the underlying process that leads to the translocation phenomenon is not determined consensually, even for simplified models [177]. Different studies of the translocation phenomenon can be categorized in two classes:

- Unbiased translocations, where the transport of a polymer through a membrane pore is a consequence of thermal fluctuations [71, 150, 157, 221].
- Biased translocations where the polymer is driven by chemical potential gradients [128, 129] or by external forces (such as electric field) [23, 94] or by mechanical manipulation using either optical or magnetic tweezers [110].

Polymer translocation was initially treated as a near-equilibrium process, and theoretically reduced to a drift-diffusion problem using the Fokker-Planck equation [126, 152, 204]. In addition, the scaling of the translocation time with the chain length is an often addressed inquiry. For unbiased translocations, the predicted average translocation time follows

$$\tau(N) \propto a^2 \frac{N^2}{D}, \tag{5.1}$$

where N is the length (or the number of monomers) of the chain, a is the length of a polymer Kuhn segment and D denotes the diffusion coefficient. The question about the choice of the proper diffusion coefficient, D, and the nature of the diffusion process, is controversial [143]. Following [204], the hydrodynamic interaction can be neglected and $D \propto N^{-1}$. This regime corresponds to the so-called «*Rouse dynamics*». Conversely, «*Zimm dynamics*» ($D \propto N^{-1/2}$) is achieved when one considers hydrodynamics. Using a fractional Fokker-Plank equation, Dubbeldam et al. [56] included the non-equilibrium characteristics, and found $\tau(N) \propto N^{-\alpha}$ with $\alpha \simeq 2.49$. This exponent is extremely close to to the prediction for *Zimm dynamics*.

On the other hand, in case of biased translocations, Lubensky and Nelson [126], Muthukumar [152] have demonstrated that the translocation time follows

$$\tau(N) \propto N^{\alpha} f^{\beta}. \tag{5.2}$$

Here, f measures the driving force. Recent theories around force driven translocations seem to agree with this relation [156]. In presence of strong driving forces, $\alpha = \beta = 1$ for

Rouse dynamics [152]. Moreover, many computer simulations have reported $1 \leq \alpha \leq 1.75$ and $\beta \simeq 1$ [94]. Later experimental studies have found various values of α (*e.g* 1.27 [199], 1.34 [63] or 1.40 [220]). Recently, Saito and Sakaue [182] have scrutinized the asymmetry in the motion of polymer segments during biased translocations. In fact, segments that have already crossed the pore are pushed out while the others are pulled into the pore. The first motion corresponds to a 3D slow diffusion while the other correspond to a 1D fast diffusion. In addition, they predict how the total driving force is dynamically allocated between the pulling and the pushing sides and showed that, in the asymptotic limit of long chains, exponents read

$$\begin{cases} \alpha &= 1+\nu \\ \beta &= p_{\nu,1} - p_z \end{cases} \tag{5.3}$$

Here, ν corresponds to the Flory exponent and $p_{\nu,d} := \nu^{-1} - d$ with d the effective dimensionality of the diffusion. $p_z := z-2$ accounts for hydrodynamics effect as $z = 2+\nu^{-1}$ in *Rouse dynamics* and $z = 3$ for *Zimm dynamics*. A similar result has been derived by Rowghanian and Grosberg [177] who introduced the iso-flux model by keeping mass flow conserved throughout the translocation. However, these scaling dependences are only valid in the asymptotic limit of long chains when local effects, such as the friction inside the pore or its deformations, can be neglected. In particular, Menais et al. [140] showed with intensive MD simulations that the scaling behavior of the mean translocation time is affected by the static size of the pore, as well as the effect of the membrane's deformations and thermal vibrations. While accounting of the influence of the pore's friction, Ikonen et al. [98] showed that the mean translocation should be written as a sum of two terms in the form

$$\langle \tau \rangle = aN^{1+\nu} + bN \tag{5.4}$$

The first term dominates for extremely long chains and the prefactor a is derived from out-of-equilibrium dynamics induced by the driving force. Nonetheless, the second term becomes significant for typical chain lengths in both experiments and computer simulations [98]. The prefactor b arises from the interactions between the polymer and the pore [184].

In the present chapter, we will focus on the biased translocations, where the drag force is induced by an external electric field. Most prior studies around this topic were conducted with MD simulations and the studied system was often described with coarse-grained models. In many cases, the simulations are conducted with an implicit solvent

method while the motion of particles is governed by Langevin dynamics [94, 120, 129]. However, explicit solvent methods, thus more expensive, are needed in order to properly deal with hydrodynamic interactions [94].

Furthermore, since a major portion of translocation simulations are performed with an external electric field as driving force a full description of a translocation process requires to take into account charge effects of translocating polyelectrolytes such as DNA or RNA [156]. However, only few studies [14, 60, 94, 137] explicitly deal ions, especially since taking long-range electrostatic interactions into account has a significant cost. Nonetheless, we believe that these simulations might benefit from incremental algorithms dedicated to electrostatics. Precisely, we expect that an adaptively restrained of the motion of the explicit solvent, will allow us to perform fast and accurate simulations of polyelectrolyte translocations including both explicit solvent and suitable electrostatics. Therefore, this approach should reduce the cost of a proper account of hydrodynamic effects (*Zimm dynamics*).

5.2 Methodology

5.2.1 Model

A traditional way to represent a polymer makes use of the bead-spring chain model [144, 214]. Here, the polymer is described as N beads connected by $N-1$ harmonic springs. Each bead has a given weight m_b and represents a monomer. In order to build a polyelectrolyte each bead carries a charge $q_b = -2e$. The polyelectrolyte is dipped in a 100^3Å TIP3P/E water box [135]. The simulation box is divided by a 8Å-thin membrane wall. This membrane is modeled with two graphene layers which are kept immobile throughout the simulation. In the z-direction, a 1nm-diameter nanopore is hollowed through each graphene layer. We connected the two pores with a channel formed with carbon atoms. The membrane divided the simulation box in two regions:

- The **cis**[1]-side, corresponds to the region where the polyelectrolyte is initially placed [157] (Left compartment on figure 5.4(1)). We denote this region by Ω_{cis}.
- The **trans**[2]-side which corresponds to the region where the polyelectrolyte will be dragged into (Right side on figure 5.4(1)). We denote this region by Ω_{trans}.

[1] Latin preposition cis for «*on this side of*».
[2] Latin preposition trans for «*accross, on the far side, beyond*».

Analogously, the pore is denoted by Ω_{pore}. In addition, an electric field $\boldsymbol{E} = -E\vec{e}_z$ is applied inside the pore in order to achieve the translocation of the polyelectrolyte. The magnitude of the field ranges from 0.5 V/Å to 16 V/Å.

Moreover, the system is neutralized with $n_c = 2N$ positively charged counterions ($q_c = e$). In addition, 128 ion pairs (Na^+,Cl^-) are added in order to mimic a saline solution. The total number of particles in the simulation box reads $N_{total} = 52514$, $N_{total} = 52562$ or $N_{total} = 52658$ when the number of monomers is $N = 16$, $N = 32$ or $N = 64$ respectively. Finally, we illustrate the evolution of a translocation process with figure 5.4.

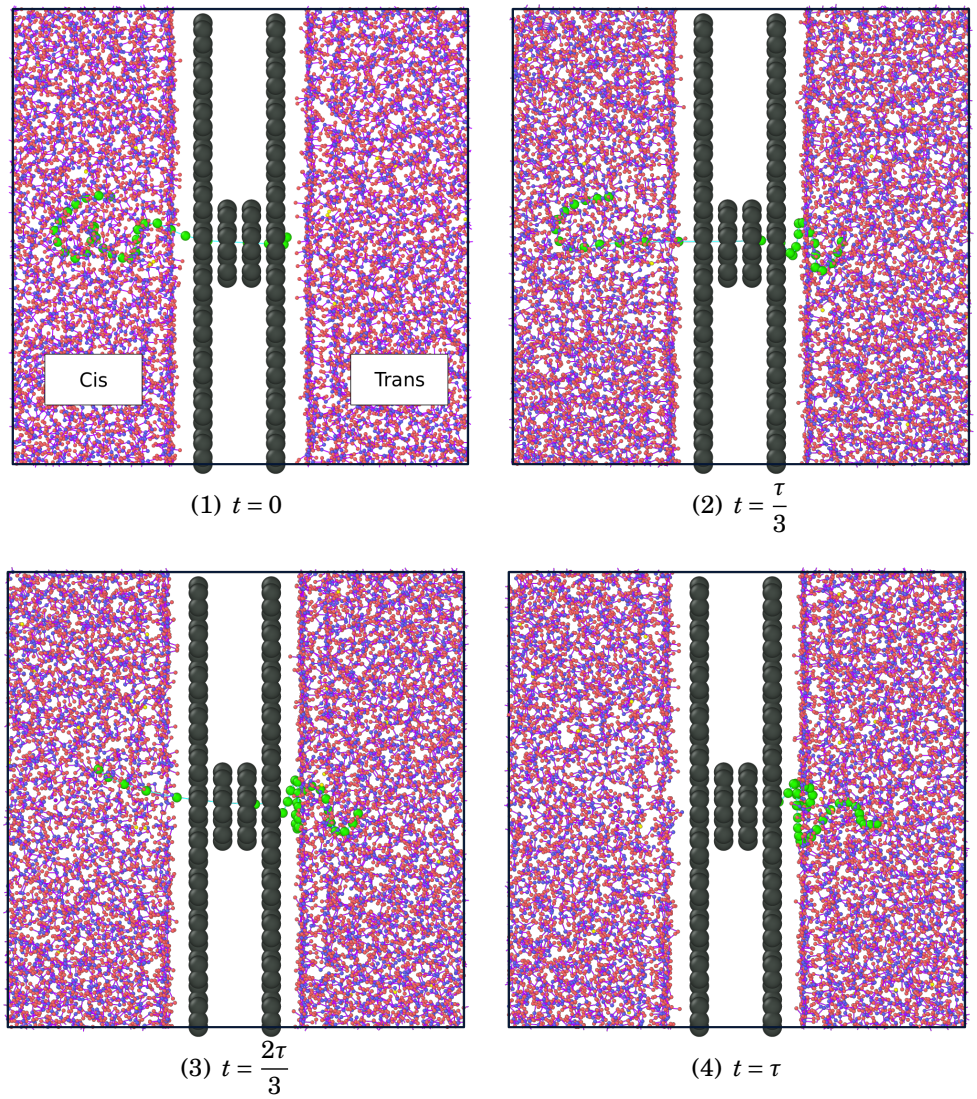


Figure 5.4: Sketch of a polyelectrolyte translocation. Monomers are shown in green. Oxygen (*resp.* Hydrogen) atoms are colored in red (blue). τ designates the translocation time.

5.2.2 Adaptively restrained Langevin dynamics

Here, the translocation process was studied with ARMD. The system was kept at the temperature $T = 300\,\mathrm{K}$ with a Langevin thermostat (*see* equation (1.30)). During the simulations, the motion of water particles were restrained with various combinations of restraining thresholds $(\varepsilon^r, \varepsilon^f)$. Carbon particles forming the membrane were kept fixed throughout the simulations. Meanwhile, other species (monomers, counterions, ions) are always active ($\varepsilon^r = 0, \varepsilon^f = 0$). The system was equilibrated for 1 ns without any external field. Then, the electric field is launched. The simulation lasted for 2 ns (including the equilibration time). The equations of motion were integrated with a timestep ranging from 0.25 fs to 1 fs depending on the magnitude of the electric field.

In order to retrieve averaged properties, we performed 100 independent simulations for each combination of parameters (namely ε_r, ε_f, N, E).

$(\varepsilon^r, \varepsilon^f)$	[%] of restrained
(0.00, 0.00)	0.0
(0.01, 0.03)	31.4
(0.10, 0.30)	73.8
(1.00, 3.00)	90.1
(5.00, 8.00)	95.7

Table 5.1: Average percentage of restrained particles throughout adaptively restrained simulations of polymer translocation $N = 32$.

5.2.3 Interaction potentials

In our model, we described pairwise interactions between particles with a repulsive Lennard-Jones potential and electrostatics.

$$V_{ij}(r_{ij}) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \tag{5.5}$$

where q_i represents the electric charge of particle i, r_{ij} is the distance between particle i and particle j, σ_{ij} represents the LJ-separation distance and ϵ_{ij} is the depth of the LJ potential well. Besides pairwise interactions, adjacent beads which form the polymer, were connected with a bond-length potential based on the Hook's law, namely,

$$V_{bond}(r) = \frac{1}{2}k\,|r - r_0|^2 \tag{5.6}$$

where r is the distance between covalently bonded monomers, r_0 defines the equilibrium bond length and k is the spring constant. Oxygen and hydrogen particles interact via a bond-angle potential and angle harmonic potential was employed.

$$V_{angle}(\theta) = \frac{1}{2} A \, |\theta - \theta_0|^2 \tag{5.7}$$

The parameters used in this research are summarized in table 5.2.

Atom	q_i (e)	σ_{ii} (Å)	ϵ_{ii} (kcal/mol)	m_i (g/mol)
M$^{(a)}$	-2.000	0.077	3.67	80.00
X$^{(b)}$	1.000	0.077	3.67	20.00
Na	1.000	2.520	0.003	23.00
Cl	-1.000	3.850	0.382	35.45
H	0.445	0.000	0.000	1.00
O	-0.890	3.188	0.169	16.00
C$^{(c)}$	0.000	3.550	0.074	12.01
Bond-length potential				
Harmonic	k (kcal/mol/Å^2)		r_0 (Å)	
M–M	5960		2.000	
O–H	450		1.098	
Bond-angle potential				
Harmonic	A (kcal/mol/°^{2})		θ_0 (°)	
H–O–H	55		104.52	

(a)M corresponds to negatively charged monomer forming the polyelectrolyte; (b)X are counter-ions; (c)C corresponds to carbon particle forming the membrane.

Table 5.2: Values of potential parameters: C corresponds to charged particle forming the nanopore and X are counter ions.

5.2.4 Incremental algorithms

In order to compare methods presented in this thesis, we handle electrostatics with the following algorithms

- P3M is our reference method. In order to achieve a 10^{-5} relative accuracy in forces, the method required a 10 Å cutoff and $75 \times 75 \times 75$ grid. The Ewald parameter was equal to $0.294\,\text{Å}^{-1}$.
- The incremental P3M was employed in two manners. First, only short-range components were incrementally updated (*see* section 3.3.1). The method was set up with aforementioned parameters. We will denote this approach by (P3M+).
- When more than 50% of particles are restrained, we allow the full incremental update of electrostatics. Then, two grids are used for $\boldsymbol{k}$-space calculations (*see* section 3.5.2). The finest mesh uses $80 \times 80 \times 80$ grid points. The size of the coarse grid is influenced by both the proportion of restrained particles and the proportion of switched particles (64^3, 66^3, 70^3 and 72^3 grid points). In practice, only one grid is used We will use (P3M++) to refer to this approach.
- We also evaluated IMCM (*see* section 4.3). In order to achieve the desired precision (10^{-5}), the method use a $64 \times 64 \times 64$ grid. Particles were assigned to the grid with a 5^{th}order B-spline charge density. The normalized cutoff of the charge density is equal to 0.12 (*i.e* 12Å cutoff for the near-field correction).

5.2.5 Additional information

These benchmarks were performed using a cluster with 8 nodes equipped with 8/16 CPUs Intel Xeon E5540 and a Gigabit Ethernet network. For each simulation, we used 8 CPUs of the same node. Furthermore, for each combination of parameters (namely ε_r, ε_f, N, E), the (100) independent simulations were reproduced four times (once for each algorithm which treats electrostatics).

5.3 Results

As electrostatics are evaluated at a similar accuracy, the averaged properties obtained in this study, are independent of the algorithms employed for computations. Therefore, we can first present several results based solely on the use of ARMD in the study of the translocation process (*see* sections 5.3.1 to 5.3.3). Then, we will compare the performance of incremental algorithms for electrostatics (*see* section 5.3.4).

5.3.1 Translocation time

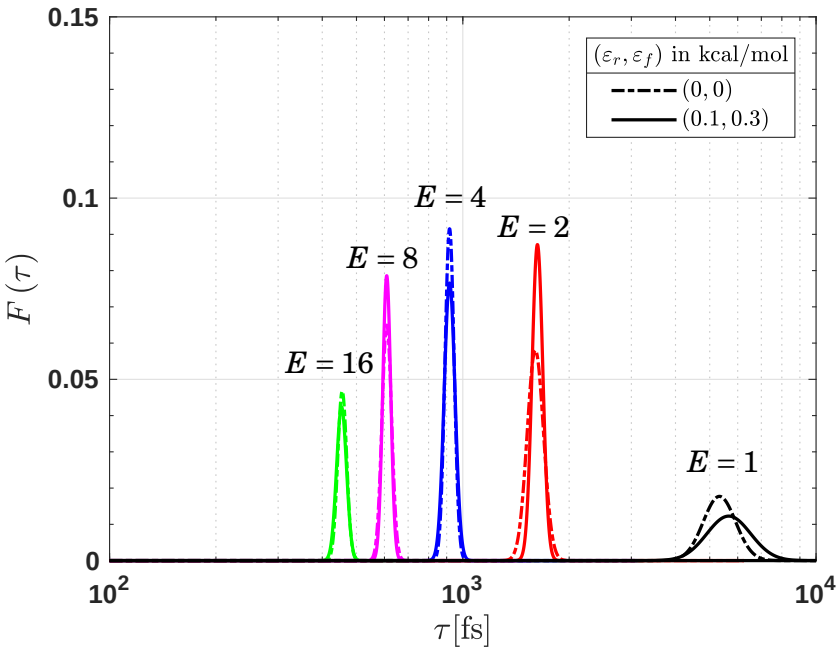


Figure 5.5: Translocation time distributions of the polymer ($N = 32$) driven by an electric field. Two sets of restraining parameters $(\varepsilon_r, \varepsilon_f)$ were compared. Distributions obtained with $(0,0)$[kcal/mol] are displayed with dash dotted line. Results from $(0.1, 0.3)$[kcal/mol] simulations are shown with solid line. Various magnitudes of the electric field E were considered. A base-10 logarithmic scale is employed for the x-axis. Raw data were fitted to the time passage distribution (equation (5.8)).

Distribution of translocation times

Panja et al. [157] defines the translocation or transit time as the average time for a polymer to navigate across a membrane. In our case, the translocation time τ corresponds to the time required for the polyelectrolyte chain to complete a passage through the pore, starting from the commissioning of the electric field. Lubensky and Nelson [126] have motivated the fact that the translocation process of a polymer passing through the

pore can be seen, in an equivalent way, as the pore undergoing one dimensional biased Brownian motion along the polymer. Therefore, the translocation process should follow a first-passage time distribution [123, 141], namely,

$$F(\tau) = \frac{N}{\sqrt{4\pi D\tau^3}} \exp\left[-\frac{(N-v\tau)^2}{4D\tau}\right]. \tag{5.8}$$

Here, N the length of the polymer chain, D the diffusion coefficient of the membrane sliding along the polymer, and v the mean translocation velocity. The reader's attention is drawn to appendix B.1 which contains some details on the derivation of equation (5.8).

Moreover, we analyzed the repartition of translocation times associated to various magnitudes of the electric field . Data were fitted to the first passage time distribution 5.8. In figure 5.5, we output, for the polymer of size N=32, the corresponding distributions obtained from ARMD simulations. As expected, the translocation time decreases with the electric field strength (E). We compared ARMD simulations obtained with ε_r = 0.1 kcal/mol and $\varepsilon_f = 0.3$ kcal/mol to the reference simulations $(\varepsilon_r, \varepsilon_f) = (0,0)$. We noticed that the associated distributions are similar, especially for strong electric fields. However, an increase is observed while reducing the strength of the field (for instance for $E = 1$ V/Å). We think that this behavior is a side effect of metastability of the system of interest.

Metastability

In molecular simulations, the dynamics can be trapped in some regions of the phase space for a long time due to energetic or entropic barriers [210]. This situation which characterizes the metastability, often corresponds to the case when the phase space possesses regions of high probability (also called metastable states) separated from each other by regions with very low probability. Thus, the transitions between high-probability regions occur rarely. The average hitting (or exit) time of a metastable state starting from another metastable state can be used as a measure of metastability.

Without any driving force, the translocation process requires the crossing of an entropic[3] barrier. The narrower the pore, the "*higher*" the barrier. In addition, a modification of the kinetic energy in the Hamiltonian of the system can increase or decrease the metastability of the dynamics [198]. In case of ARMD, when the number of active particles diminishes, the metastability may increase, which in turn induces an augmentation of the time needed to cross a given barrier. Moreover, the height of the barrier is reduced

[3]Conversely, an energetic barrier is induced by the energy landscape defined by the interaction potentials. This landscape may contains local minima or "valleys" separated by "mountains" which limit the transition between valleys.

by the application of the external field. This must explain why translocation times are similar for large magnitudes of E.

Scaling of the mean translocation time

Furthermore, we monitored the mean translocation time $\langle\tau\rangle$, obtained from ARMD simulations for various chain lengths and force strengths. figure 5.6 displays $\langle\tau\rangle$ as function of the applied electric field for various chain lengths (16, 32 and 64). Although their mean translocation time may slightly differ for low driving forces, ARMD simulations showed similar scaling laws between the mean translocation time and the magnitude E. In order to retrieve exact averages for low magnitudes of E, one may re-weight translocation times with respect to Girsanov weights. This approach has been suggested by Trstanova [210] in order to retrieve exact dynamical properties (*e.g.* exit times, transport properties) which are modified by ARMD. Earlier, this strategy has been explored by Chodera et al. [42] in order to improve the estimation of dynamical properties from simulations at multiple temperatures (*e.g.* from simulated [134] or parallel [84] tempering simulations). Nevertheless, we did not feel the need to embrace this approach for an accurate reading of the laws of power.

For weak electric fields ($E \leq 1$V/Å), the mean translocation time scales as $E^{-1.8}$ when $N = 32$. A similar behavior is observed for the polymer of size $N = 64$. This dependence testifies to the narrowness of the nanopore studied here, as it seems to agree with the work of Menais et al. [140] on the polymer translocation through vibrating thin membranes. They observed that, for narrow pores, the relation 5.2 ($\langle\tau\rangle \propto f^{\beta}$) has two main regimes. $-2 \leq \beta < -1$ was reported for weak driving forces while $\beta \simeq -1$, corresponding to the characteristic exponent in Rouse dynamics, was obtained for strong forces. Menais et al. [140] argued that a narrow pore introduces additional steric effects, summing up in an increased friction presented by the pore which slows down the translocation process for low forces.

Interestingly, in our case, the scaling law reads $\langle\tau\rangle \propto E^{-0.4}$, for large magnitudes of the electric field. This behavior is indeed expected since it reflects a transition towards Zimm dynamics. As we made use of explicit solvent, hydrodynamics interactions slow significantly the translocation process, even for large driving forces.

Moreover, when $N = 16$, the polyelectrolyte responds almost linearly to weak driving forces ($\langle\tau\rangle \propto E^{-1.2}$ when $0.5 \leq E \leq 2$). We think that this polymer is less affected by the relative narrowness of the pore. As the chain size is reduced, fewer monomers attempt to penetrate the pore, leading to a less clogged entry. Therefore, the polyelectrolyte needs

to deform less in order to perform the translocation. Although the pore is narrow, the translocation process is facilitated.

At this point, we would like to stress that scaling laws we have found for the electric field magnitude, does not exactly give the force exponent β (although close to it). Unlike other strategies (*e.g.* tensile forces applied to the end of the polymer), the electrical force acting in the pore is not constant throughout the translocation. The magnitude of the applied force depends on the number of particles located in the pore. An insight of the exact force applied is given by monitoring the average number of monomers located in the pore. From this information, the reader can accurately extract the value of β.

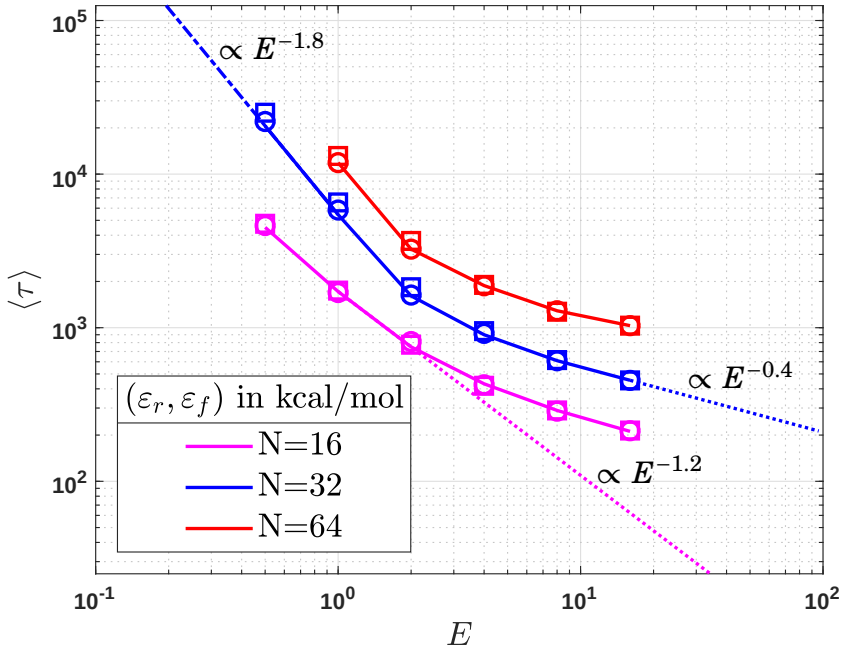


Figure 5.6: Mean translocation time $\langle\tau\rangle$ as function of the magnitude E of the electric field for chain length $N = 16$ (magenta), $N = 32$ (blue) and $N = 64$ (red)). Various configurations of restraining parameters $(\varepsilon_r, \varepsilon_f)$ were evaluated. Precisely, $(0,0)$[kcal/mol] is represented with solid line, $(0.1,0.3)$[kcal/mol] with spheres, $(1.0,3.0)$ with squares.

5.3.2 Translocation Coordinate

The variations of the number of monomers in the three compartments of the simulation box are often scrutinized in the literature [56, 94, 143]. We denote by N_{pore}, the number of monomers in the pore Ω_{pore}. Analogously, N_{cis} and N_{trans} count respectively the number of monomers in cis- and trans-regions. In particular, N_{trans} is usually denoted as the «*translocation coordinate*» [55, 56]. One may monitor the progress of the translocation

with the help of N_{trans}, since it indicates the number of monomers that successfully performed the process.

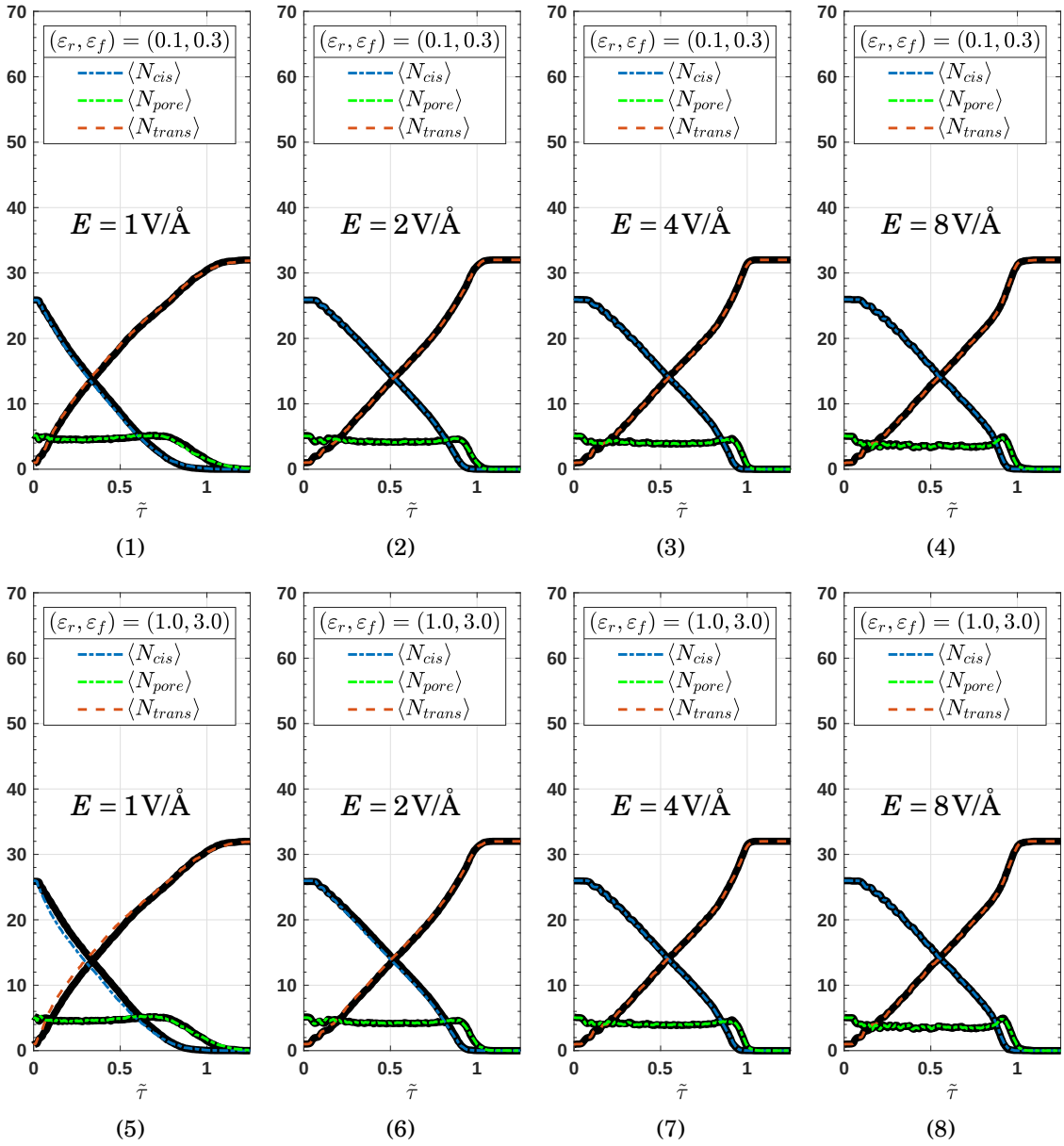


Figure 5.7: Variations of the number of monomers in the cis-region (blue), trans-region (red) and the pore (green) for the $N = 32$ polymer. Various configurations of restraining parameters $(\varepsilon_r, \varepsilon_f)$ are compared. Precisely, figures (1-4) output results froms simulations where $(\varepsilon_r, \varepsilon_f) = (0.1, 0.3)$[kcal/mol]. Analogously bottom figures (5-8) are associated to $(\varepsilon_r, \varepsilon_f) = (1.0, 3.0)$[kcal/mol]. The results from reference simulations ($\varepsilon_r = \varepsilon_f = 0$ kcal/mol) are plotted with thick black solid lines and are available in figure B.2.

Nevertheless, Hsiao [94] showed that, due to the diffusion characteristics of the translocation phenomenon, N_{trans} often shows a highly fluctuating evolution. Consequently, one must investigate the averaged translocation coordinate $\langle N_{trans} \rangle$ in order to extract the drift-diffusion behavior of the polymer.

$\langle N_{trans} \rangle$ monotonically increases from 1 to N during the threading process while the average number of monomers in the pore region remains almost constant during the translocation. Moreover, $\langle N_{pore} \rangle$ is independent of the polyelectrolyte's size. During a translocation process, the number of monomers located in the pore, is almost constant. Consequently, $\langle N_{cis} \rangle$, the average number of monomers in the cis-region, gradually decreases to zero, since $N_{trans} + N_{pore} + N_{cis} = N$.

Indeed, $\langle N_{pore} \rangle$ is essentially influenced by the length of the channel to be crossed, by the stiffness of the bonds linking monomers together and by the strength of the driving force. Under the action of electric force, the polymer is stretched. While increasing the electric field, the average distance between monomers in the pore is increased. N_{pore} is thus reduced.

In figure 5.7, we compare the average value of the translocation coordinate obtained from some ARMD simulations. We find that the progression of those simulations is independent of the chosen restraints. Moreover, we observe that the translocation progress relatively more in advance for $E = 1\,\mathrm{V/Å}$. For this magnitude of electric field, the average number of monomers located in the cis-compartment is the highest.

5.3.3 Chain Size and Positions of Chain Ends

A polymer's wingspan is often described by its radius of gyration R_g. Given $\boldsymbol{r}_i$ the positions of monomers, the radius of gyration reads

$$R_g = \frac{1}{N} \left(\sum_{i=1}^{n} (\boldsymbol{r}_i - \bar{\boldsymbol{r}})^2 \right)^{1/2} \tag{5.9}$$

where $\bar{\boldsymbol{r}} = \frac{1}{N} \sum_{i=1}^{n} \boldsymbol{r}_i$, stands for the mean position of the monomers. Moreover, R_g alongside with the end-to-end distance gives an estimation of the polymer's size. This property is of interest since it can be determined experimentally with the help of static light scattering as well as small angle neutron- and X-ray scattering.

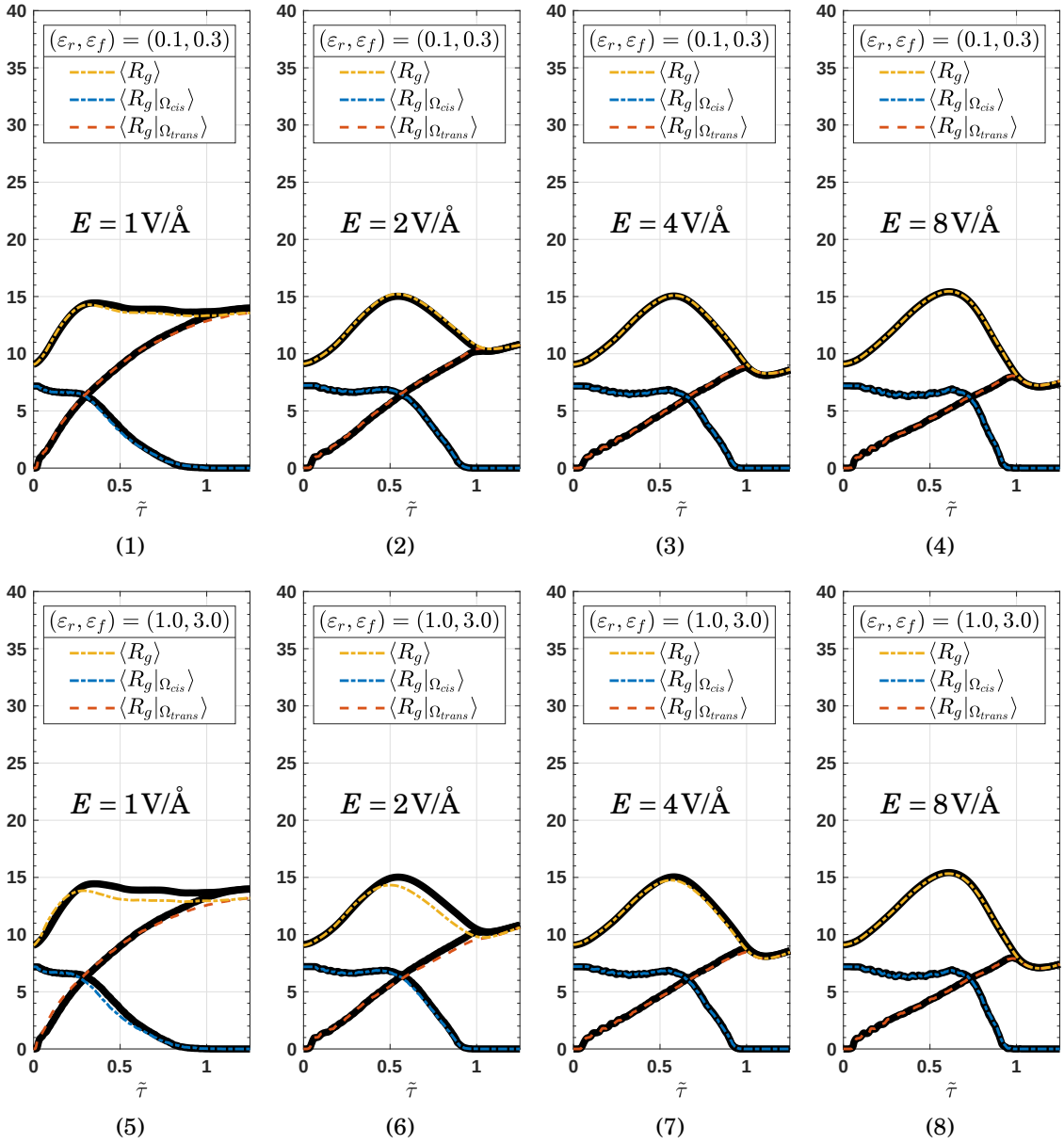


Figure 5.8: Variations of the averaged gyration for the polymer ($N = 32$) in the cis-side (blue), trans-side (red), whole domain (orange). Various configurations of restraining parameters $(\varepsilon_r, \varepsilon_f)$ are compared. Figures (1-4) output results froms simulations where $(\varepsilon_r, \varepsilon_f) = (0.1, 0.3)$[kcal/mol]. Analogously bottom figures (5-8) are associated to $(\varepsilon_r, \varepsilon_f) = (1.0, 3.0)$[kcal/mol]. The results from reference simulations ($\varepsilon_r = \varepsilon_f = 0$kcal/mol) are plotted with thick black solid lines and are available in figure B.3.

Here, we extend the definition of the radius of gyration by introducing its restriction to Ω_A , a given sub-space of the simulation box. Basically, we define the radius of gyration of the monomers that fall inside Ω_A

$$R_g|_{\Omega_A} = \frac{1}{N_A}\left(\sum_{i=1;\boldsymbol{r}_i\in\Omega_A}^{N}\left(\boldsymbol{r}_i - \bar{\boldsymbol{r}}|_{\Omega_A}\right)^2\right)^{1/2} \tag{5.10}$$

where, N_a designates the number of monomers located inside Ω_A. In addition, $\bar{\boldsymbol{r}}|_{\Omega_A}$ denotes the center of mass of monomers located inside Ω_A. We analyzed the radii of gyration associated to monomers present in cis- and trans- regions. In particular, we scrutinized the variations of their value throughout the translocation process.

Figure 5.8 outputs, for some restraining parameters, the evolution of $\langle R_g|_{\Omega_{cis}}\rangle$ and $\langle R_g|_{\Omega_{trans}}\rangle$ as a function of the normalized translocation time $\tilde{\tau}$. The radius of gyration of the whole polymer, $\langle R_g\rangle$, is also plotted in figure 5.8.

As the threading progresses, the polymer size in the cis-side diminishes, which induces a decay of $\langle R_g|_{\Omega_{cis}}\rangle$. This decay can be broken down into two phases. First of all, a plateau is observed for the first moments of the translocation. The length of the plateau increases with the magnitude of the magnitude of the driving force. Moreover, the plateau is followed by a rapid decrease of $\langle R_g|_{\Omega_{cis}}\rangle$. On the other hand, $\langle R_g|_{\Omega_{trans}}\rangle$ increases monotonically until all monomers have successfully translocated. Meanwhile, $\langle R_g\rangle$ displays a hump whose crest coincides with the instant where $\langle N_{cis}\rangle$ and $\langle N_{trans}\rangle$ evenly match. In the early stages of the translocation, the polyelectrolyte is stretched by the driving force and its size grows. At the middle of the process, the polyelectrolyte forms a dumbbell-like structure, leading to the formation of the hump [94, 157].

Additionally, the hump appears to be more pronounced as the driving force increases. More interestingly, when the electric field is strong enough, we notice, at the very end of the translocation, a compression of the polymer which results in a reduction of the chain size in the trans region $\langle R_g|trans\rangle$. However, after the compression, the polymer keeps growing. Hsiao [94] has explained this effect as a combination of two effects:

- The releasing of chain from a compressed state
- The diffusion of chain from a location near the wall to the bulk solution.

While comparing results obtained from ARMD simulations, we discovered that the evolution of these radii of gyration are extremely similar. While increasing the number of restrained particles, a slight divergence is observed in the evolution of $\langle R_g|_{\Omega_{trans}}\rangle$ for weak

driving forces ($E \leq 2\,\text{V/Å}$). This discrepancy is more significant as the polyelectrolyte approaches the end of the translocation. Interestingly, $\langle R_g|_{\Omega_{cis}} \rangle$ seems to be less affected by a reduction of the number of active particles. Even for strongly restrained simulations, the differences against results obtained with full dynamics simulations are barely perceptible.

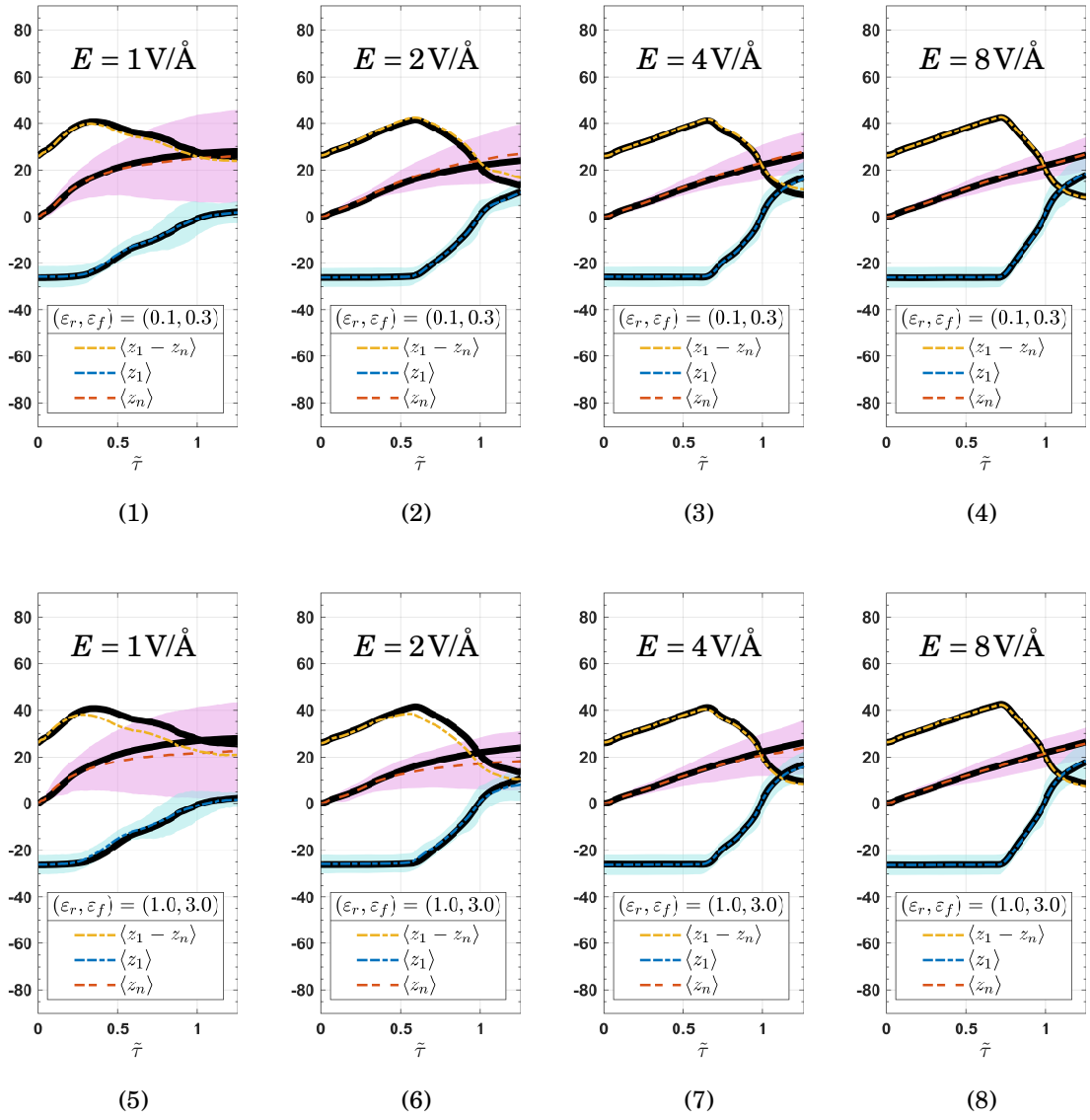


Figure 5.9: Variations of the averaged z-coordinates of the chain ends ($\langle z_1 \rangle$, $\langle z_N \rangle$) and the difference $\langle z_1 - z_N \rangle$ for the polymer ($N = 32$). Various configurations of restraining parameters ($\varepsilon_r, \varepsilon_f$) are compared. The results from reference simulations ($\varepsilon_r = \varepsilon_f = 0\,\text{kcal/mol}$) are plotted with thick black solid lines and are available in figure B.4. The $3\sigma-$ distribution range of $\langle z_N \rangle$ and $\langle z_1 \rangle$ is denoted by a surrounding colored region.

In spite of adaptive restrains, the motions of monomers in the cis-region are well-preserved. In order to confirm this allegation, we monitor the motion (in z-dimension) of the polymer ends. Here, z_1 refers to the z-coordinate of the heading monomer. Analogously, z_n corresponds to the z-coordinate of the terminal monomer. We believe that z_1 is representative of the motion of monomers in the trans-region, since the heading bead is the monomer which spends the most time in this compartment. In a similar way, the evolution of z_n will help us understand the motion of monomers in the cis-region. Figure 5.9 outputs the evolution of the averaged value of z_1 and z_n throughout the translocation for some ARMD simulations where the chain length is equal to 32. The 3σ -distribution range of the z_1 (*resp.* z_n), is estimated with the standard deviation and represented as a cyan (*resp.* pink)-colored region in figure 5.9. We also analyzed the variations of the mean difference $\langle z_1 - z_n \rangle$ as a function of $\tilde{t}$.

$\langle z_1 \rangle$ increases continuously during the translocation. Surprisingly, at the end of the process, the value of the averaged z-coordinate of the chain head seems to be independent of the magnitude of the driving force. However, an increase of the electric field E results in a decrease of the range of the distribution of z_1. Furthermore, the standard deviation of z_1 increases as the translocation progresses. Meanwhile, the z-coordinate of the terminal monomer remains constant in the first part of the translocation. For a relevant period of time, this monomer is not influenced instantly by the threading [94]. This period where the value of z_n only depends on its (negative) initial value, is more or less long depending on the magnitude of the driving force.

When $E = 8\,\mathrm{V/Å}$, the last monomer stays close of its initial z-position for almost 75% of the translocation time. Then, this monomer is pulled and threaded through the pore. During this period where $\langle z_n \rangle$ increases to zero, the fluctuations of z_n decrease drastically as the magnetic field intensifies. The motion of the monomer is constrained by the tensile force and by the pore. This leads to a funnel-shaped structure in the time evolution the 3σ -distribution range of the z_n. Upon completion of the translocation, the standard deviation of z_n keep increasing since the last monomer diffuses in the trans-region. Moreover, the time evolution of $\langle z_1 - z_n \rangle$ can be broken down into two successive trends. First, $\langle z_1 - z_n \rangle$ increases since the heading end is pushed away from the pore. Then, $\langle z_1 - z_n \rangle$ decreases as the terminal monomer is pulled. The transition between these two phases occurs later when the electric field is increased.

While comparing results from ARMD simulations, we noticed that the averaged motion in z-direction of the terminal monomer is well preserved. On the other hand, $\langle z_1 \rangle$ is perfectly recovered for strong driving forces. When the electric field is weak, a

slight divergence arises in the mean value of z_1 for strongly restrained simulations. Interestingly, this divergence occurs when the mean z-coordinate of the last monomer starts increasing.

5.3.4 Performance of incremental algorithms

Let us now compare the computational cost of the employed incremental algorithms. To do so, we compute the relative speedup of each method with respect to the reference P3M. Let denote by $T[P3M]$ the total CPU time required by all simulations performed using a given couple of restraining parameters $(\varepsilon_r, \varepsilon_f)$ and P3M. Analogously, we define $T[P3M(+)]$, $T[P3M(++)]$ and $T[IMCM]$. For a given couple of restraining parameters $(\varepsilon_r, \varepsilon_f)$, the relative speedup is computed by normalizing these timings by $T[P3M]$.

In figure 5.10, we plot these ratios against the average number of restrained particles. Incremental P3M(PM) is the most efficient algorithm when the number of restrained particles is below 80%. The method achieves a $\times 2$ when $\simeq 75\%$ of particles are restrained. When less than 20% are active, IMCM seems to be the best choice for the incremental update of electrostatics. The method was nearly 4 times faster than the traditional P3M when $\simeq 96\%$ of particles are restrained. of particles are restrained.

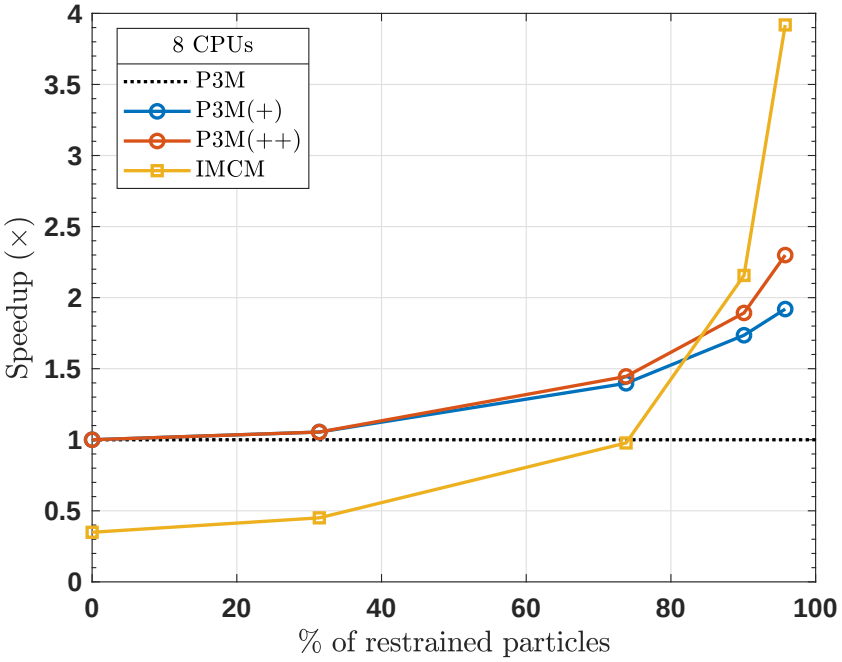


Figure 5.10: Performance of incremental algorithms on translocation simulations.

5.4 Conclusion

In this chapter, we showed that ARMD can be used to understand the motion of polymers during a translocation process. Precisely, we performed explicit water simulations of the threading of a polyelectrolyte through a narrow pore. Although, we adaptively restrained water molecules, we were able to retrieve several properties of the translocation. The other challenge of the study is to accelerate these simulations with the help of incremental algorithms specialized for electrostatics. As shown in the previous section, we successfully reduced the computational cost of the simulations. The ideal algorithm for incremental update of electrostatics may combine both incremental MCM and incremental P3M. While the latter is effective for almost all percentages of restrained particles, IMCM is the method of choice for low ratios of active particles.

CHAPTER 6

CONCLUSION AND OUTLOOK

In this thesis, we have introduced several incremental algorithms which enable an efficient calculation of long-range interactions throughout molecular simulations. These algorithms have demonstrated their effectiveness on various particle systems whose dynamics were governed by ARMD. Since ARMD switches positional degrees of freedom on and off during a simulation, the proposed algorithms may rapidly compute long-range interactions, most notably electrostatics.

- In chapter 3, we have shown that although Fourier-based methods are widely adopted for evaluating electrostatics, they can be improved in order to capitalize on adaptive restraints. First, we enhanced several stages of the Ewald summation. Then, we proposed a generic approach which incrementally updates electrostatics while using mesh-based Ewald methods. The method was presented for P3M but an extension to analogous methods is straightforward. The incremental P3M reduces the cost of electrostatics, provided that the proportion of particles which switch from restrained to active is kept low at each timestep.

- Since we felt that the proposed method did not deliver ultimate performances (especially on systems with a low number of active particles), we introduced in chapter 4 the Incremental Meshed Continuum Method. We showed (on several adaptively restrained simulations) that meshed continuum method outperforms P3M when a large number of particles is restrained. Moreover, incremental MCM and incremental P3M are complementary techniques for electrostatics in ARMD

simulations.

- ◎ In chapter 5, we have simulated the threading of a polyelectrolyte with the help of ARMD. We showed that ARMD still recovers some useful statistics. In addition, the simulations were performed faster by taking advantage of incremental methods for electrostatics. IMCM was efficient when a large portion of the solvent is restrained. However, incremental P3M is more appealing in other cases. An ideal strategy for treating electrostatics may combine both methods.

Due to lack of time, many ideas and applications have been left for the future. Therefore, several directions may be of particular interest in future studies. These studies may comprise deeper analysis of proposed algorithms, new proposals to try different methods and applications.

- We want to broaden the panoply of incremental methods for electrostatics. In particular, an incremental Fast Multipole Method should be considered. However, an accurate comparison of this approach may require an efficient implementation. Other methods such as Multilevel Summation Method might also be improved. In particular, the short-range terms of MSM can directly benefit from incremental methods dedicated to short-range potentials.

- Moreover, we want to extend the incremental P3M to *mesh staggering* or *interlacing* methods which reduce aliasing errors [38]. Furthermore, the incremental P3M that we proposed subsume that the two sub-problems are solved using the same order for charge assignment functions. It would be possible to review our incremental P3M to handle the case of a more general nature.

- We also want to enhance the incremental MCM by analyzing various theoretical aspects, most notably the control of the error, the choice of the smooth charge density.

- We have shown that we can study coulombic systems where statistics are computed only for a small subset of particles. In view of the good results obtained in this manuscript, we believe that ARMD can be extended to many other applications. Among these applications might be channels in membrane proteins, ion implantation, molecular docking, protein folding...

- In order to understand the motions that are most fundamental to the activity of a protein, we want to investigate if the correlated motions of proteins are well preserved by ARMD [7].
- Moreover, we expect that the results of chapter 5 can be extended to more general cases (*e.g.* the polymer translocation through nanopores in vibrating thin membranes [140]). In addition, we may investigate the effect of adaptive restraints on the averaged properties in ARMD simulations of unbiased translocations.
- In this work, we used at most 16 CPU cores. A direct extension for the near future is the use of more powerful multi-core environments in order to improve the parallelization. In particular, we want to evaluate our incremental algorithms on parallel GPU computing architectures.
- Finally, since our incremental methods are not confined to molecular systems. We may evaluate their performance on systems where gravitational interactions dominate.

Part IV

Appendices

APPENDIX A

ERROR ESTIMATION IN P3M

A.1 Scaling of the RMS error in (partial) forces

Here, we address the dependence of the Root Mean Square (RMS) error in forces on the number of charged particles and their valence.

Let us consider a collection $\mathbb{L}_0$ of N charges q_i located in a simulation box. We examine the *partial* electrostatic forces $\boldsymbol{f}^{(\mathbb{L})}$ induced by a subset $(\mathbb{L})$ of the collection of particles. The size of $(\mathbb{L})$ is denoted as $N_{\mathbb{L}}$. The Root Mean Square (RMS) error in forces is defined as

$$\Delta f^{(\mathbb{L})} := \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\Delta \boldsymbol{f}_i^{(\mathbb{L})}\right)^2} \tag{A.1}$$

$$:= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{f}_i^{(\mathbb{L})} - \boldsymbol{f}_{i,ref}^{(\mathbb{L})}\right)^2} \tag{A.2}$$

where $\boldsymbol{f}_i^{(\mathbb{L})}$ is the force on particle i induced by the influence of the subset $(\mathbb{L})$ of particles. $\boldsymbol{f}_{i,ref}^{(\mathbb{L})}$ is the exact force on that particle.

For a given particle i, the error in $\boldsymbol{f}_i^{(\mathbb{L})}$ originates from each pairwise interaction of i with a particle j of the set $\mathbb{L} \setminus \{i\}$. Since each interaction is proportional to the product of the valences of particles involved, we assume that the corresponding error is as well proportional to the product of the valences.

Therefore, $\Delta \boldsymbol{f}_i^{(\mathbb{L})}$ reads

$$\Delta \boldsymbol{f}_i^{(\mathbb{L})} = q_i \sum_{j \in \mathbb{L} \setminus i} q_j \boldsymbol{\chi}_{ij} \tag{A.3}$$

Here, $\boldsymbol{\chi}_{i,j}$ is the error in forces between two unit charges ($q = 1$) placed at location of i and j. At this point, we can assume that contributions $\boldsymbol{\chi}_{i,j}$ from different particles j are uncorrelated. We acknowledge that this assumption is certainly not valid for highly ordered particle distributions or strongly inhomogeneous systems. Under this assumption, the average of the scalar product $\boldsymbol{\chi}_{ij}.\boldsymbol{\chi}_{ik}$ over the particle configurations reads

$$\left\langle \boldsymbol{\chi}_{ij}.\boldsymbol{\chi}_{ik} \right\rangle = \delta_{jk} \left\langle \boldsymbol{\chi}_{ij}^2 \right\rangle \tag{A.4}$$

The reader should note that $\langle \boldsymbol{\chi}_{ij}^2 \rangle$, the mean square force error for two unit charges, is independent on i and j. Hence, $\delta_{jk} \left\langle \boldsymbol{\chi}_{ij}^2 \right\rangle =: \delta_{jk} \chi^2$. From equations (A.3) and (A.4), the mean square force on i reads

$$\begin{aligned} \left\langle \left(\Delta \boldsymbol{f}_i^{(\mathbb{L})} \right)^2 \right\rangle &= \left\langle \left(q_i \sum_{j \in \mathbb{L} \setminus \{i\}} q_j \boldsymbol{\chi} ik \right) . \left(q_i \sum_{k \in \mathbb{L} \setminus \{i\}} q_k \boldsymbol{\chi} ij \right) \right\rangle \\ &= q_i^2 \sum_{j \in \mathbb{L} \setminus \{i\}} \sum_{k \in \mathbb{L} \setminus \{i\}} \langle \boldsymbol{\chi} ij.\boldsymbol{\chi} ik \rangle \\ &\simeq q_i^2 \chi^2 Q_{\mathbb{L}}^2 \end{aligned} \tag{A.5}$$

where $Q_{\mathbb{L}}^2$ is the total square sum of charges in the subset $\mathbb{L}$, namely,

$$Q_{\mathbb{L}}^2 := \sum_{i \in \mathbb{L}} q_i^2 \tag{A.6}$$

At this point, we assume that let there are P subsets $\mathbb{N}_p$ of particle sharing the same valence θ_p. If $|\mathbb{N}_p| \gg 1$, the law of large numbers stipulates that, for each particle $i \in \mathbb{N}_p$, the arithmetic mean of $\left(\Delta \boldsymbol{f}_i^{(\mathbb{L})} \right)^2$ can be approximated by the ensemble average for one particle of $\mathbb{N}_p$, namely

$$\frac{1}{|\mathbb{N}_p|} \sum_{i \in \mathbb{N}_p} \left(\Delta \boldsymbol{f}_i^{(\mathbb{L})} \right)^2 \simeq \left\langle \left(\Delta \boldsymbol{f}_i^{(\mathbb{L})} \right)^2 \right\rangle . \tag{A.7}$$

Thus,

$$\frac{1}{|\mathbb{N}_p|} \sum_{i \in \mathbb{N}_p} \left(\Delta \boldsymbol{f}_i^{(\mathbb{L})} \right)^2 \simeq \theta_p \chi^2 Q_{\mathbb{L}}^2 . \tag{A.8}$$

Assuming all $\mathbb{N}_p$ large, one can write

$$\frac{1}{N}\left(\Delta \boldsymbol{f}_i^{(\mathbb{L})}\right)^2 = \sum_{p=1}^{P} \frac{|\mathbb{N}_p|}{N}\left(\frac{1}{|\mathbb{N}_p|}\sum_{i\in\mathbb{N}_p}\left(\Delta \boldsymbol{f}_i^{(\mathbb{L})}\right)^2\right) \tag{A.9}$$

$$\simeq \frac{\chi^2 Q_{\mathbb{L}}^2}{N}\sum_{p=1}^{P}|\mathbb{N}_p|\theta_p$$

$$\simeq \frac{\chi^2 Q_{\mathbb{L}}^2 Q_{\mathbb{L}_0}^2}{N} \tag{A.10}$$

which implies,

$$\Delta \boldsymbol{f}^{(\mathbb{L})} \simeq \chi\sqrt{\frac{Q_{\mathbb{L}}^2 Q_{\mathbb{L}_0}^2}{N}}. \tag{A.11}$$

Here, $Q_{\mathbb{L}_0}^2$ denotes the total square sum of all charges. Finally, let us assume that the particle system is homogeneous. This assumption implies

$$Q_{\mathbb{L}_0}^2 \simeq \frac{N_{\mathbb{L}}}{N}Q^2 \tag{A.12}$$

Consequently, the RMS error in *partial* forces – induced by charges in $\mathbb{L}$ – scales as

$$\Delta \boldsymbol{f}^{(\mathbb{L})} \simeq \chi\frac{Q_{\mathbb{L}}^2}{\sqrt{N_{\mathbb{L}}}}. \tag{A.13}$$

A.2 Error measure in P3M

In [90], $\mathscr{F}$ denoted the displacement-averaged total squared deviation of $(\boldsymbol{F}(\boldsymbol{x}_2-\boldsymbol{x}_1,\boldsymbol{x}_1))$, the mesh-based approximation of the k-space contribution of the force between two unit charges located at positions $\boldsymbol{x}_1$, $\boldsymbol{x}_2=\boldsymbol{x}_1+\boldsymbol{x}$, from a reference inter-particle force $\boldsymbol{F}_{\boldsymbol{r}}(\boldsymbol{x}_2-\boldsymbol{x}_1)$. Basically,

$$\mathscr{F} := \frac{1}{h^3}\int_{h^3} \mathrm{d}^3x_1 \int_{L^3} \mathrm{d}^3x\,[\boldsymbol{F}(\boldsymbol{x},\boldsymbol{x}_1)-\boldsymbol{F}_{\boldsymbol{r}}(\boldsymbol{x})]^2. \tag{A.14}$$

As explained in [90], a particular choice of the charge assignment function, implies $\mathscr{F}$ is essentially a quadratic functional of the influence function G.

For P3M, the used influence function G_{opt} is chosen in order to minimize $\mathscr{F}$.

Hockney and Eastwood [90] employs the i$\boldsymbol{k}$-differentiation in their original derivation of P3M. In this case, the optimal influence function reads (in Fourier space)

$$\tilde{G}_{opt}(\boldsymbol{k}) = \frac{\tilde{D}(\boldsymbol{k}).\sum_{m\in\mathbb{Z}^3}\tilde{U}(\boldsymbol{k}+2\pi\boldsymbol{m}/h)\tilde{\boldsymbol{F}}_r(\boldsymbol{k}+2\pi\boldsymbol{m}/h)}{|\tilde{D}(\boldsymbol{k})|^2\left[\sum_{m\in\mathbb{Z}^3}\tilde{U}(\boldsymbol{k}+2\pi\boldsymbol{m}/h)\right]^2} \tag{A.15}$$

where

$$\tilde{\boldsymbol{F}}_r(\boldsymbol{k}) = -i\boldsymbol{k}\frac{4\pi}{k^2}e^{-\sigma^2k^2/2}. \tag{A.16}$$

$$\tilde{U}(\boldsymbol{k}) = \tilde{W}(\boldsymbol{k})/h^3. \tag{A.17}$$

Here $\tilde{D}(\boldsymbol{k})$ is the Fourier transform of the chosen differential operator. In case of ik-P3M, $\tilde{D}(\boldsymbol{k}) = i\boldsymbol{k}$. $\tilde{W}$ denotes the Fourier transform of the chosen assignment function, namely,

$$\tilde{W}(\boldsymbol{k}) = h^3\left[\frac{\sin\left(\frac{1}{2}k_xh\right)}{\frac{1}{2}k_xh}\frac{\sin\left(\frac{1}{2}k_yh\right)}{\frac{1}{2}k_yh}\frac{\sin\left(\frac{1}{2}k_zh\right)}{\frac{1}{2}k_zh}\right]^P \tag{A.18}$$

which corresponds to the Fourier transform of a spline of order P with a compact support $\left[-\frac{Ph}{2},\frac{Ph}{2}\right]^3$ [53].

The corresponding value of the functional give the «*optimal error*» which reads

$$\mathscr{F}[G_{opt}] = \frac{1}{L^3}\sum_{\boldsymbol{m}\in\mathbb{Z}^3}\left[|\tilde{\boldsymbol{R}}|(\boldsymbol{k}+2\pi\boldsymbol{m}/h)|^2 - \frac{\tilde{D}(\boldsymbol{k}).\sum_{m\in\mathbb{Z}^3}\tilde{U}(\boldsymbol{k}+2\pi\boldsymbol{m}/h)\tilde{\boldsymbol{F}}_r{}^*(\boldsymbol{k}+2\pi\boldsymbol{m}/h)|^2}{|\tilde{D}(\boldsymbol{k})|^2\left[\sum_{m\in\mathbb{Z}^3}\tilde{U}(\boldsymbol{k}+2\pi\boldsymbol{m}/h)\right]^2}\right] \tag{A.19}$$

Here, $\tilde{\boldsymbol{F}}_r{}^*$ denotes the complex conjugate of $\tilde{\boldsymbol{F}}_r$ which has been introduced in (A.16). $\tilde{D}(\boldsymbol{k})$ refers to the Fourier transform of the chosen differential operator, which corresponds to $i\boldsymbol{k}$ in case of ik-P3M. In the above expression, $\tilde{U}$ depends of the choice of the charge assignment function.

A.3 Expansion coefficients $a_m^{(P)}$ used in (3.48)

In this thesis, we restricted ourselves to the analytical approximation of $\mathscr{F}[G_{opt}]$ derived by Deserno and Holm [53], namely (equation (3.48)),

$$\Delta f \simeq \frac{Q^2(h\beta)^P}{\sqrt{N}L^2}\sqrt{\sqrt{2}\pi\beta L\sum_{m=0}^{P-1}a_m^{(P)}(h\beta)^m}$$

The following table contains the value of $(a_m^{(P)})$ employed in [53], for the analytical approximation of the ik-P3M optimal error $\mathscr{F}[\tilde{G}_{opt}]$ (equation (3.48)) .

P	a_0^P	a_1^P	a_2^P	a_3^P	a_4^P	a_5^P	a_6^P
1	$\frac{2}{3}$						
2	$\frac{1}{50}$	$\frac{5}{294}$					
3	$\frac{1}{588}$	$\frac{7}{1440}$	$\frac{21}{3872}$				
4	$\frac{1}{4320}$	$\frac{3}{1936}$	$\frac{7601}{2271360}$	$\frac{143}{28800}$			
5	$\frac{1}{23232}$	$\frac{7601}{13628160}$	$\frac{143}{69120}$	$\frac{517231}{106536960}$	$\frac{106640677}{11737571328}$		
6	$\frac{691}{68140800}$	$\frac{13}{57600}$	$\frac{47021}{35512320}$	$\frac{9694607}{2095994880}$	$\frac{733191589}{59609088000}$	$\frac{326190917}{11700633600}$	
7	$\frac{1}{345600}$	$\frac{3617}{35512320}$	$\frac{745739}{838397952}$	$\frac{56399353}{12773376000}$	$\frac{25091609}{1560084480}$	$\frac{1755948832039}{36229939200000}$	$\frac{4887769399}{37838389248}$

Table A.1: Expansion coefficients $a_m^{(P)}$ from Equation (3.48)

APPENDIX B

ON THE POLYMER TRANSLOCATION

B.1 Distribution of translocation times

Ling and Ling [123] proposed to study the distribution of translocation times with with Schrödinger's first-passage-time theory. In their analysis, the translocation is considered as a biased diffusion in 1-D. In addition, they consider the pore thin enough compared to the length of the polymer to be considered a walker that moves along it. Earlier, Lubensky and Nelson [126] have motivated the fact that the translocation process of a polymer passing through the pore can be seen, in an equivalent way, as the pore undergoing one dimensional biased Brownian motion along the polymer. Therefore, the motion of the polymer can be described by solving the Fokker-Planck's equation with suitable boundary conditions:

$$\frac{\partial P(x,t)}{\partial t} = D\frac{\partial^2 P(x,t)}{\partial x^2} - v\frac{\partial P(x,t)}{\partial x}. \tag{B.1}$$

where, $P(x,t)$ denotes the probability for the pore to be located at the position x of the polymer at the instant t.

Here D denotes the diffusion constant and v the drift velocity of the segment of the polymer inside the pore. At $t = 0$, they assumed that the first monomer is located inside the pore.

$$P(x,0) = \delta(x) \tag{B.2}$$

In addition, they adopted an absorbing boundary condition once the polymer has completely translocated into the trans side of the pore. Therefore, the translocated polymer cannot travel back into the pore. Given a polymer of length N, this condition reads

$$P(N,t) = 0 \tag{B.3}$$

With these conditions, the solution of the Fokker–Planck equation follows

$$P(x,t) = \frac{1}{\sqrt{4\pi D t}}\left[\exp\left(-\frac{x-vt}{4Dt}\right) - \exp\left(\frac{vN}{D}\right)\exp\left(-\frac{(-x-2N-vt)^2}{4Dt}\right)\right]. \tag{B.4}$$

The distribution of translocation times is derived with a first-passage probability density:

$$F(\tau) = \frac{\mathrm{d}}{\mathrm{d}\tau}\int_{-\infty}^{L} P(x,\tau)\mathrm{d}\tau \tag{B.5}$$

$$= \frac{N}{\sqrt{4\pi D\tau^3}}\exp\left[-\frac{(N-v\tau)^2}{4D\tau}\right]. \tag{B.6}$$

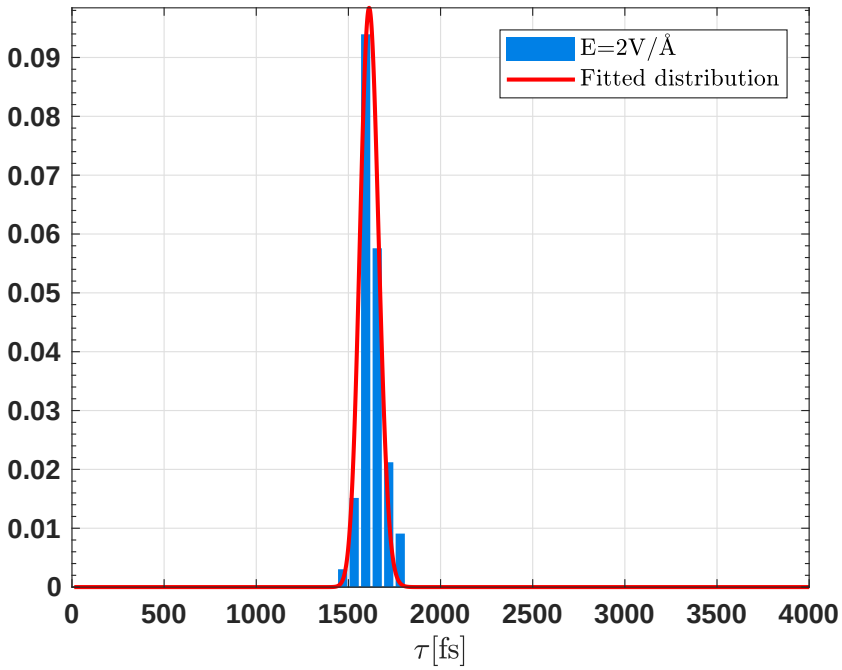


Figure B.1: Distribution of translocation times for the polymer $N = 32$ threading to a narrow pore. We also show as solid red line the fitted to first-passage probability density distribution.

B.2 Reference simulations of the translocation of a polymer (N=32)

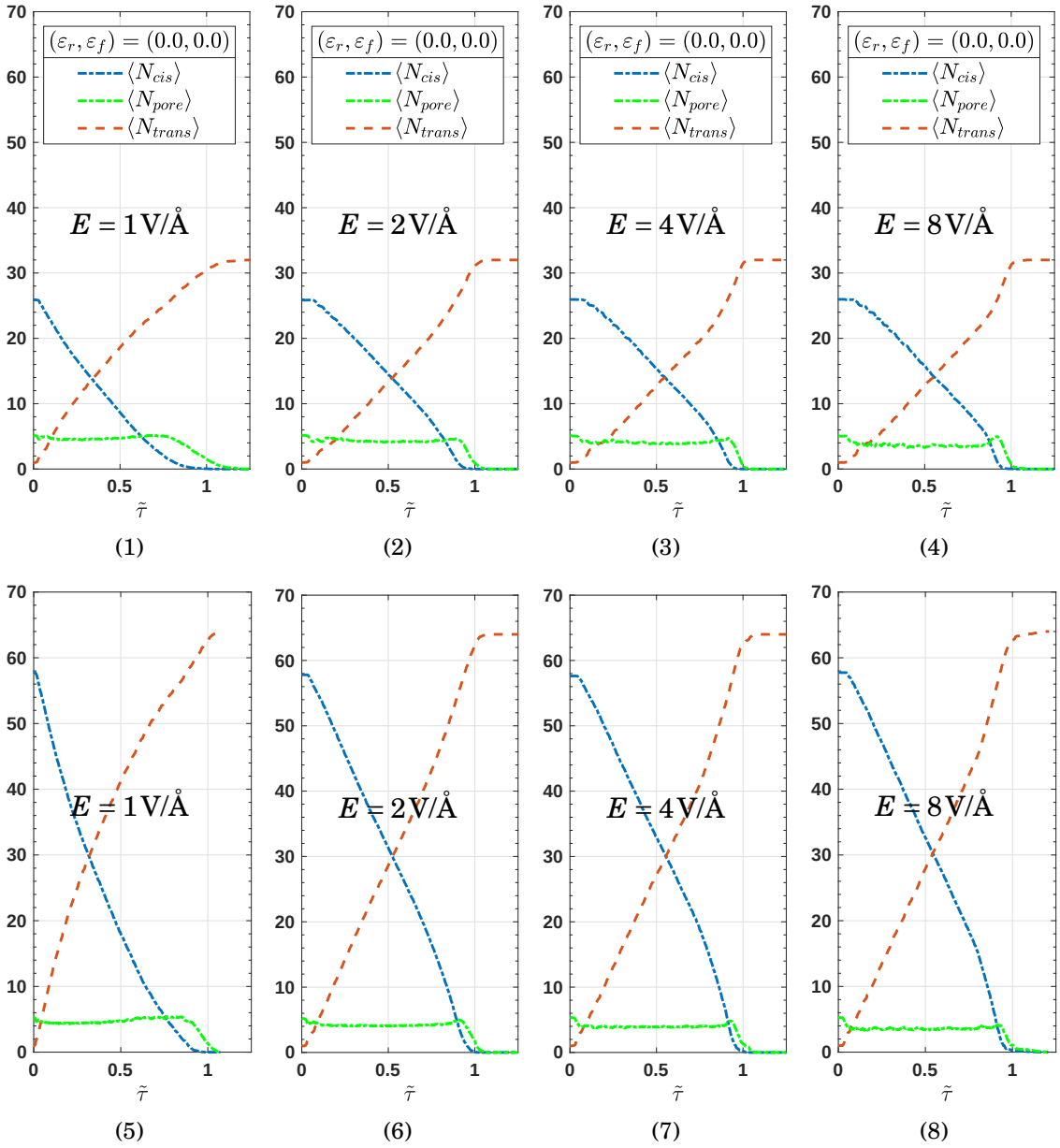


Figure B.2: Reference variations of the number of monomers in the cis-region (blue), trans-region (red) and the pore (green) for the polymer with $N = 32$ and $N = 64$. (1-4) corresponds to results for $N = 32$. (5-8) corresponds to results for $N = 64$. $\tilde{\tau}$ is the normalized translocation time.

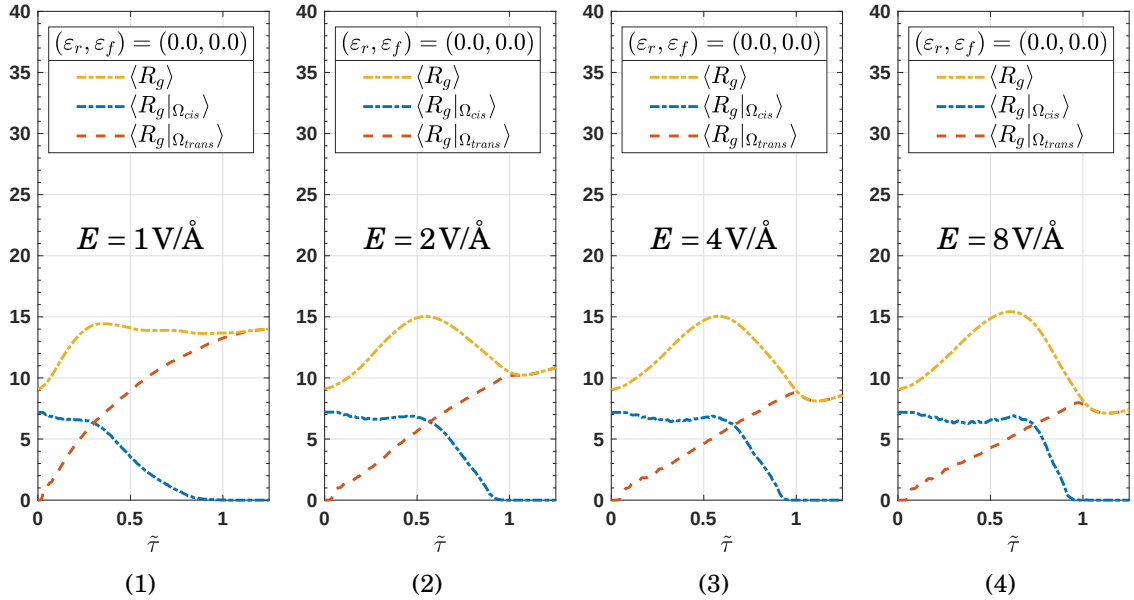


Figure B.3: Reference variations of the averaged gyration for the polymer ($N = 32$) in the cis-side (blue), trans-side (red), whole domain (orange)–– ($\varepsilon_r = \varepsilon_f = 0$kcal/mol) .

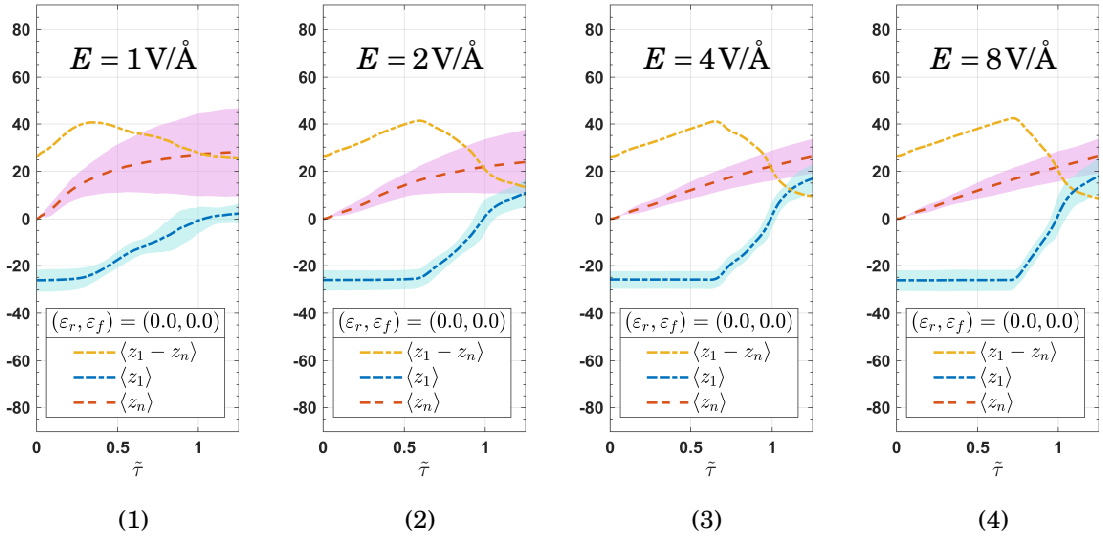


Figure B.4: Reference variations of the averaged z-coordinates of the chain ends ($\langle z_1 \rangle$, $\langle z_N \rangle$) and the difference $\langle z_1 - z_N \rangle$ for the polymer ($N = 32$) –– ($\varepsilon_r = \varepsilon_f = 0$kcal/mol).

Acronyms

ARMD Adaptively Restrained Molecular Dynamics

CPL concentration polarization layer

DCA Discretization Coarse Grid Approximation

DFT discrete Fourier Transform

FFT fast Fourier Transform

FMM Fast Multipole Method

GCA Galerkin Coarse Grid Approximation

IMCM Incremental Meshed Continuum Method

LAMMPS Large-scale Atomic/Molecular Massively Parallel Simulator

LJ Lennard-Jones

MCM meshed continuum method

MD Molecular Dynamics

MPI Message Passing Interface

MSM Multilevel Summation Method

P3M Particle Particle Particle Mesh

p.b.c periodic boundary conditions

PDE partial differential equation

PME Particle Mesh Ewald

r.h.s. right-hand side

RMS Root Mean Square

SPME Smoothed Particle Mesh Ewald

Bibliography

[1] Ruben Abagyan, Maxim Totrov, and Dmitry Kuznetsov. Icm—a new method for protein modeling and design: applications to docking and structure prediction from the distorted native conformation. *Journal of computational chemistry*, 15(5): 488–506, 1994. [p.20]

[2] S Aboud, D Marreiro, Marco Saraniti, and R Eisenberg. A poisson p 3 m force field scheme for particle-based simulations of ionic liquids. *Journal of Computational Electronics*, 3(2):117–133, 2004. [p.27]

[3] Emmanuel Agullo, Bérenger Bramas, Olivier Coulaud, Eric Darve, Matthias Messner, and Toru Takahashi. Task-based fmm for multicore architectures. *SIAM Journal on Scientific Computing*, 36(1):C66–C93, 2014. [p.27]

[4] Aleksei Aksimentiev. Deciphering ionic current signatures of dna transport through a nanopore. *Nanoscale*, 2(4):468–483, 2010. [p.126]

[5] Aleksij Aksimentiev, Jiunn B Heng, Gregory Timp, and Klaus Schulten. Microscopic kinetics of dna translocation through synthetic nanopores. *Biophysical journal*, 87(3):2086–2097, 2004. [p.126]

[6] Michael P Allen and Dominic J Tildesley. *Computer simulation of liquids*. Oxford university press, 2017. [p.61]

[7] Andrea Amadei, Antonius Linssen, and Herman JC Berendsen. Essential dynamics of proteins. *Proteins: Structure, Function, and Bioinformatics*, 17(4): 412–425, 1993. [p.149]

[8] U. Ananthakrishnaiah, Ram P Manohar, and John W Stephenson. Fourth-order finite difference methods for three-dimensional general linear elliptic problems with variable coefficients. *Numerical Methods for Partial Differential Equations*, 3 (3):229–240, 1987. ISSN 0749-159X. doi: 10.1002/num.1690030307. URL http://doi.wiley.com/10.1002/num.1690030307. [p.50], [p.100]

[9] Kravtsov Andrey and Klypin Anatoly. Formation of the large-scale structure in the universe: filaments, 2018. URL http://cosmicweb.uchicago.edu/filaments.html. [p.4]

[10] Axel Arnold, Florian Fahrenberger, Christian Holm, Olaf Lenz, Matthias Bolten, Holger Dachsel, Rene Halver, Ivo Kabadshow, Franz Gähler, Frederik Heber, Julian Iseringhausen, Michael Hofmann, Michael Pippig, Daniel Potts, and Godehard Sutmann. Comparison of scalable fast methods for long-range interactions. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 88(6): 063308, dec 2013. ISSN 15393755. doi: 10.1103/PhysRevE.88.063308. URL https://link.aps.org/doi/10.1103/PhysRevE.88.063308. [p.5], [p.10], [p.26], [p.27], [p.28], [p.50], [p.51], [p.60], [p.96], [p.97], [p.98], [p.99], [p.113]

[11] Vladimir Igorevich Arnol'd. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013. [p.7]

[12] Svetlana Artemova. *Adaptive algorithms for molecular simulation*. PhD thesis, Université de Grenoble, 2012. [p.33]

[13] Svetlana Artemova and Stephane Redon. Adaptively restrained particle simulations. *Physical Review Letters*, 109(19):190201, nov 2012. ISSN 00319007. doi: 10.1103/PhysRevLett.109.190201. URL https://link.aps.org/doi/10.1103/PhysRevLett.109.190201. [p.29], [p.30], [p.31], [p.32], [p.122]

[14] Stanislav M Avdoshenko, Daijiro Nozaki, Claudia Gomes da Rocha, Jhon W Gonzalez, Myeong H Lee, Rafael Gutierrez, and Gianaurelio Cuniberti. Dynamic and electronic transport properties of dna translocation through graphene nanopores. *Nano letters*, 13(5):1969–1976, 2013. [p.130]

[15] Perla Balbuena and Jorge M Seminario. *Molecular dynamics: from classical to quantum methods*, volume 7. Elsevier, 1999. [p.4]

[16] V. Ballenegger, J. J. Cerda, O. Lenz, and Ch Holm. The optimal P3M algorithm for computing electrostatic energies in periodic systems. *Journal of Chemical Physics*, 128(3):034109, jan 2008. ISSN 00219606. doi: 10.1063/1.2816570. URL http://aip.scitation.org/doi/10.1063/1.2816570. [p.27]

[17] Vincent Ballenegger, Jan Josep Cerdà, and Christian Holm. Removal of spurious self-interactions in particle–mesh methods. *Computer Physics Communications*, 182(9):1919–1923, 2011. [p.27], [p.73], [p.74], [p.112]

[18] Herman JC Berendsen, James PM Postma, Wilfred F van Gunsteren, and Jan Hermans. Interaction models for water in relation to protein hydration. In *Intermolecular forces*, pages 331–342. Springer, 1981. [p.22]

[19] Herman JC Berendsen, David van der Spoel, and Rudi van Drunen. Gromacs: a message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91(1-3):43–56, 1995. [p.11]

[20] HJC Berendsen, JR Grigera, and TP Straatsma. The missing term in effective pair potentials. *Journal of Physical Chemistry*, 91(24):6269–6271, 1987. [p.22]

[21] Bruce J Berne, Michal Borkovec, and John E Straub. Classical and modern methods in reaction rate theory. *The Journal of Physical Chemistry*, 92(13):3711–3725, 1988. [p.15]

[22] Sergey M Bezrukov, Igor Vodyanoy, and V Adrian Parsegian. Counting polymers moving through a single ion channel. *Nature*, 370(6487):279, 1994. [p.126]

[23] A Bhattacharya. A. bhattacharya, wh morrison, k. luo, t. ala-nissila, sc ying, a. milchev, and k. binder, eur. phys. j. e 29, 423 (2009). *Eur. Phys. J. E*, 29:423, 2009. [p.128]

[24] Leo Bluestein. A linear filtering approach to the computation of discrete fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4):451–455, 1970. [p.46]

[25] Matthias Bolten. *Multigrid methods for structured grids and their application in particle simulation*. PhD thesis, Wuppertal, Univ., Diss., 2008, 2008. [p.28], [p.40], [p.41], [p.97], [p.98]

[26] Nawaf Bou-Rabee. Time integrators for molecular dynamics. *Entropy*, 16(1): 138–162, 2013. [p.19]

[27] Ronald Newbold Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986. [p.45]

[28] James H Bramble, Richard E Ewing, Joseph E Pasciak, and Jian Shen. The analysis of multigrid algorithms for cell centered finite difference methods. *Advances in Computational Mathematics*, 5(1):15–29, 1996. [p.55]

[29] A Brandt, J Bernholc, and K Binder. *Multiscale computational methods in chemistry and physics*. NATO Science Series. IOS Press, 2001. ISBN 1-58603-141-4. URL https://books.google.fr/books?id=OzlZZn8zcCUC. [p.49]

[30] Achi Brandt and Oren E. Livne. *Multigrid Techniques : 1984 Guide With Applications to Fluid Dynamics*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2008. ISBN 1611970741. URL https://books.google.fr/books?id=EvClMX03MyAC. [p.49], [p.100]

[31] Daniel Branton, David W Deamer, Andre Marziali, Hagan Bayley, Steven A Benner, Thomas Butler, Massimiliano Di Ventra, Slaven Garaj, Andrew Hibbs, Xiaohua Huang, et al. The potential and challenges of nanopore sequencing. *Nature biotechnology*, 26(10):1146, 2008. [p.126]

[32] Donald W Brenner. Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films. *Physical review B*, 42(15):9458, 1990. [p.19], [p.20]

[33] W L Briggs, Van Emden Henson, and S F McCormick. *A Multigrid Tutorial*, volume 37. Society for Industrial and Applied Mathematics, 2000. ISBN 0898714621. doi: 10.1007/BF00046565. URL http://books.google.de/books?id=oSTGBm64o1AC. [p.27], [p.49], [p.50], [p.54], [p.55], [p.107]

[34] E Oran Brigham. *The fast Fourier transform and its applications*, volume 1. prentice Hall Englewood Cliffs, NJ, 1988. [p.45]

[35] Bernard R Brooks, Robert E Bruccoleri, Barry D Olafson, David J States, S Swaminathan, and Martin Karplus. Charmm: a program for macromolecular energy, minimization, and dynamics calculations. *Journal of computational chemistry*, 4(2):187–217, 1983. [p.11]

[36] Richard A Buckingham. The classical equation of state of gaseous helium, neon and argon. *Proc. R. Soc. Lond. A*, 168(933):264–283, 1938. [p.16]

[37] Eric Cances, Mireille Defranceschi, Werner Kutzelnigg, Claude Le Bris, and Yvon Maday. Computational quantum chemistry: a primer. *Handbook of numerical analysis*, 10:3–270, 2003. [p.11], [p.15]

[38] David S Cerutti and David A Case. Multi-level ewald: A hybrid multigrid/fast fourier transform approach to the electrostatic particle-mesh problem. *Journal of chemical theory and computation*, 6(2):443–458, 2009. [p.27], [p.148]

[39] Aparna Chandramowlishwaran, Kamesh Madduri, and Richard Vuduc. Diagnosis, tuning, and redesign for multicore performance: A case study of the fast multipole method. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12. IEEE Computer Society, 2010. [p.27]

[40] Jianhan Chen, Charles L Brooks III, and Jana Khandogin. Recent advances in implicit solvent-based methods for biomolecular simulations. *Current opinion in structural biology*, 18(2):140–148, 2008. [p.22]

[41] RCY Chin, GW Hedstrom, and FA Howes. Considerations on solving problems with multiple scales. In *Multiple Time Scales*, pages 1–27. Elsevier, 1985. [p.26]

[42] John D Chodera, William C Swope, Frank Noé, Jan-Hendrik Prinz, Michael R Shirts, and Vijay S Pande. Dynamical reweighting: Improved estimates of dynamical properties from simulations at multiple temperatures. *The Journal of chemical physics*, 134(24):06B612, 2011. [p.137]

[43] Edmond Chow and Raymond Hu. A survey of parallelization techniques for multigrid. *Parallel processing for scientific computing*, 20:179, 2006. [p.107]

[44] Charles K Chui. *Wavelets: a mathematical tool for signal analysis*, volume 1. Siam, 1997. [p.41], [p.98]

[45] Charles K Chui. *An introduction to wavelets*. Elsevier, 2016. [p.41], [p.98]

[46] G Andres Cisneros, Volodymyr Babin, and Celeste Sagui. Electrostatics interactions in classical simulations. In *Biomolecular Simulations*, pages 243–270. Springer, 2013. [p.27]

[47] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965. [p.46]

[48] Wendy D Cornell, Piotr Cieplak, Christopher I Bayly, Ian R Gould, Kenneth M Merz, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins,

nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995. [p.16]

[49] Olivier Coulaud, Pierre Fortin, and Jean Roman. Hybrid mpi-thread parallelization of the fast multipole method. In *Parallel and Distributed Computing, 2007. ISPDC'07. Sixth International Symposium on*, pages 52–52. IEEE, 2007. [p.27]

[50] Olivier Coulaud, Pierre Fortin, and Jean Roman. High performance blas formulation of the adaptive fast multipole method. *Mathematical and Computer Modelling*, 51(3-4):177–188, 2010. [p.27]

[51] Christopher J Cramer. *Essentials of computational chemistry: theories and models*. John Wiley & Sons, 2013. [p.12]

[52] D. Frenkel and B. Smit. *Understanding Molecular Simulation: from Algorithms to Applications*. Academic Press, 2002. ISBN 9780080519982. [p.5], [p.6], [p.10], [p.18], [p.19], [p.23], [p.25], [p.61]

[53] M Deserno and C Holm. How to mesh up {E}wald sums. 2. An accurate error estimate for the P3M algorithm. *J. Chem. Phys.*, 109:7694, 1998. URL http://dx.doi.org/10.1063/1.477415. [p.64], [p.73], [p.80], [p.81], [p.156]

[54] Markus Deserno and Christian Holm. How to mesh up Ewald sums. I. a theoretical and numerical comparison of various particle mesh routines. *Journal of Chemical Physics*, 109(18):7678–7693, nov 1998. ISSN 00219606. doi: 10.1063/1.477414. URL http://aip.scitation.org/doi/10.1063/1.477414. [p.26], [p.27], [p.61], [p.73], [p.74], [p.80], [p.101]

[55] JLA Dubbeldam, A Milchev, VG Rostiashvili, and Thomas A Vilgis. Polymer translocation through a nanopore: A showcase of anomalous diffusion. *Physical Review E*, 76(1):010801, 2007. [p.138]

[56] JLA Dubbeldam, Andrey Milchev, VG Rostiashvili, and Thomas A Vilgis. Driven polymer translocation through a nanopore: A manifestation of anomalous diffusion. *EPL (Europhysics Letters)*, 79(1):18002, 2007. [p.128], [p.138]

[57] Semeho Prince A. Edorh and Stéphane Redon. Incremental update of electrostatic interactions in adaptively restrained particle simulations. *Journal*

of Computational Chemistry, 0(0), 2018. doi: 10.1002/jcc.25215. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.25215. [p.35]

[58] U Essmann. U. essmann, l. perera, ml berkowitz, t. darden, h. lee, and lg pedersen, j. chem. phys. 103, 8577 (1995). *J. Chem. Phys.*, 103:8577, 1995. [p.60], [p.74]

[59] Ulrich Essmann, Lalith Perera, Max L Berkowitz, Tom Darden, Hsing Lee, and Lee G Pedersen. A smooth particle mesh Ewald method. *The Journal of Chemical Physics*, 103(19):8577–8593, nov 1995. ISSN 0021-9606. doi: 10.1063/1.470117. URL http://aip.scitation.org/doi/10.1063/1.470117. [p.26], [p.27], [p.60], [p.74]

[60] Farnoush Farahpour, Azadeh Maleknejad, Fathollah Varnik, and Mohammad Reza Ejtehadi. Chain deformation in translocation phenomena. *Soft Matter*, 9(9):2750–2759, 2013. [p.130]

[61] Scott E Feller, Richard W Pastor, Atipat Rojnuckarin, Stephen Bogusz, and Bernard R Brooks. Effect of electrostatic force truncation on interfacial and transport properties of water. *The Journal of Physical Chemistry*, 100(42):17011–17020, 1996. [p.26]

[62] Christopher J Fennell and J Daniel Gezelter. Is the ewald summation still necessary? pairwise alternatives to the accepted standard for long-range electrostatics. *The Journal of chemical physics*, 124(23):234104, 2006. [p.28]

[63] Daniel Fologea, Eric Brandin, James Uplinger, Daniel Branton, and Jiali Li. Dna conformation and base number simultaneously determined in a nanopore. *Electrophoresis*, 28(18):3186–3192, 2007. [p.129]

[64] James B Foresman, Todd A Keith, Kenneth B Wiberg, John Snoonian, and Michael J Frisch. Solvent effects. 5. influence of cavity shape, truncation of electrostatics, and electron correlation on ab initio reaction field calculations. *The Journal of Physical Chemistry*, 100(40):16098–16104, 1996. [p.26]

[65] Etienne Forest. Geometric integration for particle accelerators. *Journal of Physics A: Mathematical and General*, 39(19):5321, 2006. [p.9]

[66] Etienne Forest and Ronald D Ruth. Fourth-order symplectic integration. *Physica D: Nonlinear Phenomena*, 43(1):105–117, 1990. [p.9]

[67] Bengt Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of computation*, 51(184):699–706, 1988. [p.112]

[68] Matteo Frigo and Steven G Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93(2):216–231, 2005. [p.46]

[69] Raúl Fuentes-Azcatl and Marcia C Barbosa. Sodium chloride, nacl/ϵ: New force field. *The Journal of Physical Chemistry B*, 120(9):2460–2470, 2016. [p.22], [p.110], [p.111]

[70] Maria Fyta. Threading dna through nanopores for biosensing applications. *Journal of physics: condensed Matter*, 27(27):273101, 2015. [p.126]

[71] Michel G Gauthier and Gary W Slater. Nondriven polymer translocation through a nanopore: Computational evidence that the escape and relaxation processes are coupled. *Physical Review E*, 79(2):021802, 2009. [p.128]

[72] Walter Gautschi. *Numerical analysis*. Springer Science & Business Media, 2011. [p.101]

[73] Matthew A Gebbie, Alexander M Smith, Howard A Dobbs, Gregory G Warr, Xavier Banquy, Markus Valtiner, Mark W Rutland, Jacob N Israelachvili, Susan Perkin, Rob Atkin, et al. Long range electrostatic forces in ionic liquids. *Chemical Communications*, 53(7):1214–1224, 2017. [p.122]

[74] Anna C Gilbert, Piotr Indyk, Mark Iwen, and Ludwig Schmidt. Recent developments in the sparse fourier transform: A compressed fourier transform for big data. *IEEE Signal Processing Magazine*, 31(5):91–100, 2014. [p.76]

[75] Herbert Goldstein. *Classical mechanics*. Pearson Education India, 2011. [p.8]

[76] Tobias Gradl, Christoph Freundl, Harald Köstler, and Ulrich Rüde. Scalable multigrid. *High Performance Computing in Science and Engineering, Garching/Munich 2007*, pages 475–483, 2009. [p.49]

[77] Leslie Greengard and Vladimir Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987. [p.27]

[78] Michael Griebel, Stephan Knapek, and Gerhard Zumbusch. Numerical simulation in molecular dynamics. numerics, algorithms, parallelization, applications, volume 5 of texts in computational science and engineering, 2007. [p.98]

[79] Helmut Grubmüller, Helmut Heller, Andreas Windemuth, and Klaus Schulten. Generalized verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular Simulation*, 6(1-3):121–142, 1991. [p.26]

[80] Bertrand Guillot. A reappraisal of what we have learnt during three decades of computer simulations on water. *Journal of Molecular Liquids*, 101(1-3):219–260, 2002. [p.22]

[81] Peter Güntert, Christian Mumenthaler, and Kurt Wüthrich. Torsion angle dynamics for nmr structure calculation with the new program dyana1. *Journal of molecular biology*, 273(1):283–298, 1997. [p.20]

[82] Wolfgang Hackbusch. *Multi-grid methods and applications*, volume 4. Springer Science & Business Media, 2013. [p.49], [p.55], [p.96]

[83] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006. [p.8], [p.9]

[84] Ulrich HE Hansmann. Parallel tempering algorithm for conformational studies of biological molecules. *Chemical Physics Letters*, 281(1-3):140–150, 1997. [p.137]

[85] David J Hardy, Zhe Wu, James C Phillips, John E Stone, Robert D Skeel, and Klaus Schulten. Multilevel Summation Method for Electrostatic Force Evaluation. *Journal of Chemical Theory and Computation*, 11(2):766–779, feb 2015. ISSN 1549-9618. doi: 10.1021/ct5009075. URL http://pubs.acs.org/doi/abs/10.1021/ct5009075. [p.27], [p.28], [p.96]

[86] David J Hardy, Matthew A Wolff, Jianlin Xia, Klaus Schulten, and Robert D Skeel. Multilevel summation with b-spline interpolation for pairwise interactions in molecular dynamics simulations. *The Journal of chemical physics*, 144(11):114112, 2016. [p.28]

[87] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Simple and practical algorithm for sparse fourier transform. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1183–1194. Society for Industrial and Applied Mathematics, 2012. [p.76]

[88] PW Hemker. On the order of prolongations and restrictions in multigrid procedures. *Journal of Computational and Applied Mathematics*, 32(3):423–429, 1990. [p.55]

[89] Hitachi. Hitachi, 2018. URL http://www.hitachi.com/rd/portal/highlight/genome/index.html. [p.127]

[90] R W Hockney and J W Eastwood. *Computer Simulation Using Particles*, volume 25. crc Press, 1988. ISBN 0852743920. doi: 10.1137/1025102. URL http://link.aip.org/link/SIREAD/v25/i3/p425/s1{&}Agg=doi. [p.25], [p.26], [p.60], [p.74], [p.75], [p.80], [p.81], [p.155]

[91] Christian Holm. Efficient methods for long range interactions in periodic geometries plus one application. *Computational Soft Matter: From Synthetic Polymers to Proteins*, 23:195–236, 2004. [p.63]

[92] William G Hoover. Canonical dynamics: equilibrium phase-space distributions. *Physical review A*, 31(3):1695, 1985. [p.6]

[93] Lars Hörmander. Hypoelliptic second order differential equations. *Acta Mathematica*, 119(1):147–171, 1967. [p.9]

[94] Pai-Yi Hsiao. Polyelectrolyte threading through a nanopore. *Polymers*, 8(3):73, 2016. [p.22], [p.128], [p.129], [p.130], [p.138], [p.140], [p.142], [p.144]

[95] Guohui Hu, Mao Mao, and Sandip Ghosal. Ion transport through a graphene nanopore. *Nanotechnology*, 23(39):395501, 2012. [p.110]

[96] Markus Hülsemann, Frankand Kowarschik, Marcus Mohr, and Ulrich Rüde. *Parallel Geometric Multigrid*, pages 165–208. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-31619-0. doi: 10.1007/3-540-31619-1_5. URL https://doi.org/10.1007/3-540-31619-1_5. [p.49], [p.96], [p.107]

[97] Nathan G Hunt and Fred E Cohen. Fast lookup tables for interatomic interactions. *Journal of computational chemistry*, 17(16):1857–1862, 1996. [p.25]

[98] Timo Ikonen, Aniket Bhattacharya, Tapio Ala-Nissila, and Wokyung Sung. Influence of pore friction on the universal aspects of driven polymer translocation. *EPL (Europhysics Letters)*, 103(3):38001, 2013. [p.129]

[99] Illumina. Illumina, 2018. URL https://www.illumina.com/. [p.127]

[100] A Jain, N Vaidehi, and G Rodriguez. A fast recursive algorithm for molecular dynamics simulation. *Journal of Computational Physics*, 106(2):258–268, 1993. [p.20]

[101] Frank Jensen. *Introduction to computational chemistry*. John wiley & sons, 2017. [p.16]

[102] William L Jorgensen and Jeffry D Madura. Quantum and statistical mechanical studies of liquids. 25. solvation and conformation of methanol in water. *Journal of the American Chemical Society*, 105(6):1407–1413, 1983. [p.22]

[103] William L Jorgensen, Jeffry D Madura, and Carol J Swenson. Optimized intermolecular potential functions for liquid hydrocarbons. *Journal of the American Chemical Society*, 106(22):6638–6646, 1984. [p.20]

[104] Wetterstrand KA. Dna sequencing costs: Data from the nhgri genome sequencing program (gsp), 2018. URL www.genome.gov/sequencingcostsdata. [p.127]

[105] Kai Kadau, Timothy C Germann, and Peter S Lomdahl. Molecular dynamics comes of age: 320 billion atom simulation on bluegene/l. *International Journal of Modern Physics C*, 17(12):1755–1761, 2006. [p.4]

[106] Parimal Kar and Michael Feig. Hybrid all-atom/coarse-grained simulations of proteins by direct coupling of charmm and primo force fields. *Journal of chemical theory and computation*, 13(11):5753–5765, 2017. [p.21]

[107] John J Kasianowicz, Eric Brandin, Daniel Branton, and David W Deamer. Characterization of individual polynucleotide molecules using a membrane channel. *Proceedings of the National Academy of Sciences*, 93(24):13770–13773, 1996. [p.126]

[108] John J Kasianowicz, Joseph WF Robertson, Elaine R Chan, Joseph E Reiner, and Vincent M Stanford. Nanoscopic porous sensors. *Annu. Rev. Anal. Chem.*, 1: 737–766, 2008. [p.126]

[109] WH Keesom. The second viral coefficient for rigid spherical molecules, whose mutual attraction is equivalent to that of a quadruplet placed at their centre. *Proc. R. Acad. Sci*, 18:636–646, 1915. [p.15]

[110] Ulrich F Keyser. Controlling molecular transport through nanopores. *Journal of The Royal Society Interface*, page rsif20110222, 2011. [p.128]

[111] Sebastian Kmiecik, Dominik Gront, Michal Kolinski, Lukasz Wieteska, Aleksandra Elzbieta Dawid, and Andrzej Kolinski. Coarse-grained protein models and their applications. *Chemical Reviews*, 116(14):7898–7936, 2016. [p.20], [p.21]

[112] Jiri Kolafa and John W Perram. Cutoff errors in the Ewald summation formulae for point charge systems. *Molecular Simulation*, 9(5):351–368, 1992. ISSN 0892-7022. doi: 10.1080/08927029208049126. URL http://www.icpf.cas.cz/jiri/papers/ewalderr/default.htm. [p.60], [p.63], [p.84]

[113] David B Kony, Wolfgang Damm, Serge Stoll, Wilfred F Van Gunsteren, and Philippe H Hünenberger. Explicit-solvent molecular dynamics simulations of the polysaccharide schizophyllan in water. *Biophysical journal*, 93(2):442–455, 2007. [p.21]

[114] Markus Kowarschik and Christian Weiss. Dimepack–a cache-optimized multigrid library. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications - VOLUME I*. Citeseer, 2001. [p.49]

[115] Ryogo Kubo, Morikazu Toda, and Natsuki Hashitsume. *Statistical physics II: nonequilibrium statistical mechanics*, volume 31. Springer Science & Business Media, 2012. [p.10]

[116] Michael Kuhlen, Mark Vogelsberger, and Raul Angulo. Numerical simulations of the dark universe: State of the art and the next decade. *Physics of the Dark Universe*, 1(1-2):50–93, 2012. [p.5]

[117] Do Y Kwak. V-cycle multigrid for cell-centered finite differences. *SIAM Journal on Scientific Computing*, 21(2):552–564, 1999. [p.54]

[118] LAMMPS. Spc/e water system (4500 particles),http://lammps.sandia.gov/bench/bench_spce.tar.gz. http://lammps.sandia.gov/bench/bench_spce.tar.gz, 2017. Accessed: 2017-09-30. [p.68]

[119] Michael Levitt and Arieh Warshel. Computer simulation of protein folding. *Nature*, 253(5494):694, 1975. [p.20]

[120] Jiapeng Li, Yan Zhang, Juekuan Yang, Kedong Bi, Zhonghua Ni, Deyu Li, and Yunfei Chen. Molecular dynamics study of dna translocation through graphene nanopores. *Physical Review E*, 87(6):062707, 2013. [p.130]

[121] Hai Lin and Donald G Truhlar. Qm/mm: what have we learned, where are we, and where do we go from here? *Theoretical Chemistry Accounts*, 117(2):185, 2007. [p.20]

[122] Zhibin Lin, Elodie Leveugle, Eduardo M Bringa, and Leonid V Zhigilei. Molecular dynamics simulation of laser melting of nanocrystalline au. *The Journal of Physical Chemistry C*, 114(12):5686–5699, 2009. [p.4]

[123] Daniel Y Ling and Xinsheng Sean Ling. On the distribution of dna translocation times in solid-state nanopores: an analysis using schrodinger s first-passage-time theory. *Journal of Physics: Condensed Matter*, 25(37):375102, 2013. [p.136], [p.159]

[124] Harvey Lodish, Arnold Berk, S Lawrence Zipursky, Paul Matsudaira, David Baltimore, James Darnell, et al. *Molecular cell biology*, volume 3. WH Freeman New York, 1995. [p.126]

[125] Base4 Innovation Ltd. Base4, 2018. URL http://www.base4.co.uk/microdroplet-sequencing/. [p.127]

[126] David K Lubensky and David R Nelson. Driven polymer translocation through a narrow pore. *Biophysical Journal*, 77(4):1824–1838, 1999. [p.128], [p.135], [p.159]

[127] Jinping Luo, Lijun Liu, Peng Su, Pengbo Duan, and Daihui Lu. A piecewise lookup table for calculating nonbonded pairwise atomic interactions. *Journal of molecular modeling*, 21(11):288, 2015. [p.25]

[128] Kaifu Luo, Tapio Ala-Nissila, See-Chen Ying, and Aniket Bhattacharya. Sequence dependence of dna translocation through a nanopore. *Physical review letters*, 100 (5):058101, 2008. [p.128]

[129] Kaifu Luo, Santtu TT Ollila, Ilkka Huopaniemi, Tapio Ala-Nissila, Pawel Pomorski, Mikko Karttunen, See-Chen Ying, and Aniket Bhattacharya. Dynamical scaling exponents for polymer translocation through a nanopore. *Physical Review E*, 78 (5):050901, 2008. [p.128], [p.130]

[130] Buyong Ma and Ruth Nussinov. Explicit and implicit water simulations of a β-hairpin peptide. *Proteins: Structure, Function, and Bioinformatics*, 37(1):73–87, 1999. [p.22]

[131] Alexander D MacKerell, Bernard Brooks, Charles L Brooks, Lennart Nilsson, Benoit Roux, Youngdo Won, and Martin Karplus. Charmm: the energy function and its parameterization. *Encyclopedia of computational chemistry*, 1998. [p.19]

[132] Alex D MacKerell Jr, Donald Bashford, MLDR Bellott, Roland Leslie Dunbrack Jr, Jeffrey D Evanseck, Martin J Field, Stefan Fischer, Jiali Gao, H Guo, Sookhee Ha, et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The journal of physical chemistry B*, 102(18):3586–3616, 1998. [p.13]

[133] Michael W Mahoney and William L Jorgensen. A five-site model for liquid water and the reproduction of the density anomaly by rigid, nonpolarizable potential functions. *The Journal of Chemical Physics*, 112(20):8910–8922, 2000. [p.22]

[134] Enzo Marinari and Giorgio Parisi. Simulated tempering: a new monte carlo scheme. *EPL (Europhysics Letters)*, 19(6):451, 1992. [p.137]

[135] Pekka Mark and Lennart Nilsson. Structure and dynamics of the tip3p, spc, and spc/e water models at 298 k. *The Journal of Physical Chemistry A*, 105(43): 9954–9960, 2001. [p.130]

[136] Siewert J Marrink, H Jelger Risselada, Serge Yefimov, D Peter Tieleman, and Alex H De Vries. The martini force field: coarse grained model for biomolecular simulations. *The journal of physical chemistry B*, 111(27):7812–7824, 2007. [p.20]

[137] Jérôme Mathé, Aleksei Aksimentiev, David R Nelson, Klaus Schulten, and Amit Meller. Orientation discrimination of single-stranded dna inside the α-hemolysin membrane channel. *Proceedings of the National Academy of Sciences of the United States of America*, 102(35):12377–12382, 2005. [p.130]

[138] Rousset Mathias, Stoltz Gabriel, and Lelievre Tony. *Free Energy Computations: A Mathematical Perspective*. World Scientific, 2010. [p.10]

[139] Stephen L Mayo, Barry D Olafson, and William A Goddard. Dreiding: a generic force field for molecular simulations. *Journal of Physical chemistry*, 94(26):8897–8909, 1990. [p.13]

[140] Timothée Menais, Stefano Mossa, and Arnaud Buhot. Polymer translocation through nano-pores in vibrating thin membranes. *Scientific reports*, 6:38558, 2016. [p.129], [p.137], [p.149]

[141] Ralf Metzler and Joseph Klafter. When translocation dynamics becomes anomalous. *Biophysical journal*, 85(4):2776, 2003. [p.136]

[142] A Michels, Hub Wijker, and HK Wijker. Isotherms of argon between 0 c and 150 c and pressures up to 2900 atmospheres. *Physica*, 15(7):627–633, 1949. [p.33]

[143] Andrey Milchev. Single-polymer dynamics under constraints: scaling theory and computer experiment. *Journal of Physics: Condensed Matter*, 23(10):103101, 2011. [p.126], [p.128], [p.138]

[144] Andrey Milchev and Kurt Binder. Static and dynamic properties of adsorbed chains at surfaces: Monte carlo simulation of a bead-spring model. *Macromolecules*, 29(1): 343–354, 1996. [p.130]

[145] Gradimir V. Milovanović and Zlatko Udovičić. Calculation of coefficients of a cardinal B-spline. *Applied Mathematics Letters*, 23(11):1346–1350, 2010. ISSN 08939659. doi: 10.1016/j.aml.2010.06.029. URL http://dblp.uni-trier.de/db/journals/appml/appml23.html{#}MilovanovicU10. [p.41], [p.98]

[146] Antonio Miotello and Paolo M Ossi. *Laser-surface interactions for new materials production*, volume 130. Springer, 2010. [p.6]

[147] Utkur Mirsaidov, Jeffrey Comer, Valentin Dimitrov, Aleksei Aksimentiev, and Gregory Timp. Slowing the translocation of double-stranded dna using a nanopore smaller than the double helix. *Nanotechnology*, 21(39):395501, 2010. [p.126]

[148] Marcus Mohr and Roman Wienands. Cell-centred multigrid revisited. *Computing and Visualization in Science*, 7(3):129–140, 2004. [p.54]

[149] F. Molitor, J. Güttinger, C. Stampfer, D. Graf, T. Ihn, and K. Ensslin. Local gating of a graphene hall bar by graphene side gates. *Phys. Rev. B*, 76:245426, Dec 2007. doi: 10.1103/PhysRevB.76.245426. URL https://link.aps.org/doi/10.1103/PhysRevB.76.245426. [p.119]

[150] Felipe Mondaini and L Moriconi. Markovian description of unbiased polymer translocation. *Physics Letters A*, 376(45):2903–2907, 2012. [p.128]

[151] Stan G Moore and Paul S Crozier. Extension and evaluation of the multilevel summation method for fast long-range electrostatics calculations. *The Journal of chemical physics*, 140(23):06B619_1, 2014. [p.28]

[152] Murugappan Muthukumar. Polymer translocation through a hole. *The Journal of Chemical Physics*, 111(22):10371–10374, 1999. [p.126], [p.128], [p.129]

[153] Isaac Newton. *The Principia: mathematical principles of natural philosophy*. Univ of California Press, 1999. [p.8]

[154] Lennart Nilsson. Efficient table lookup without inverse square roots for calculation of pair wise atomic interactions in classical simulations. *Journal of computational chemistry*, 30(9):1490–1498, 2009. [p.25]

[155] WG Noid. Perspective: Coarse-grained models for biomolecular systems. *The Journal of chemical physics*, 139(9):09B201_1, 2013. [p.20]

[156] Vladimir V Palyulin, Tapio Ala-Nissila, and Ralf Metzler. Polymer translocation: the first two decades and the recent diversification. *Soft matter*, 10(45):9016–9037, 2014. [p.128], [p.130]

[157] Debabrata Panja, Gerard T Barkema, and Anatoly B Kolomeisky. Through the eye of the needle: recent advances in understanding biopolymer translocation. *Journal of Physics: Condensed Matter*, 25(41):413101, 2013. [p.126], [p.128], [p.130], [p.135], [p.142]

[158] Eric Paquet and Herna L Viktor. Molecular dynamics, monte carlo simulations, and langevin dynamics: a computational review. *BioMed research international*, 2015, 2015. [p.6]

[159] Jong Hyuk Park and Andreas Heyden. Solving the equations of motion for mixed atomistic and coarse-grained systems. *Molecular Simulation*, 35(10-11):962–973, 2009. [p.21]

[160] Igor Pasichnyk and Burkhard Duenweg. Coulomb interactions via local dynamics: a molecular-dynamics algorithm. *Journal of Physics: Condensed Matter*, 16(38): S3999–S4020, sep 2004. ISSN 0953-8984. doi: 10.1088/0953-8984/16/38/017. URL http://stacks.iop.org/0953-8984/16/S3999{%}5Cnhttp://arxiv.org/abs/cond-mat/0406223. [p.28]

[161] Michael Patra, Mikko Karttunen, Marja T Hyvönen, E Falck, Peter Lindqvist, and Ilpo Vattulainen. Molecular dynamics simulations of lipid bilayers: major artifacts due to truncating electrostatic interactions. *Biophysical journal*, 84(6):3636–3645, 2003. [p.26]

[162] Juan R Perilla and Klaus Schulten. Physical properties of the hiv-1 capsid from all-atom molecular dynamics simulations. *Nature communications*, 8:15959, 2017. [p.21]

[163] Henrik G Petersen. Accuracy and efficiency of the particle mesh ewald method. *The Journal of chemical physics*, 103(9):3668–3679, 1995. [p.63]

[164] James C Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D Skeel, Laxmikant Kale, and Klaus Schulten. Scalable molecular dynamics with namd. *Journal of computational chemistry*, 26(16):1781–1802, 2005. [p.10], [p.11]

[165] Stefano Piana, John L Klepeis, and David E Shaw. Assessing the accuracy of physical models used in protein-folding simulations: quantitative evidence from long molecular dynamics simulations. *Current opinion in structural biology*, 24: 98–105, 2014. [p.18]

[166] Levi CT Pierce, Romelia Salomon-Ferrer, Cesar Augusto F. de Oliveira, J Andrew McCammon, and Ross C Walker. Routine access to millisecond time scale events with accelerated molecular dynamics. *Journal of chemical theory and computation*, 8(9):2997–3002, 2012. [p.18]

[167] S Plimpton. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics*, 117(1):1–19, 1995. ISSN 00219991. doi: 10.1006/jcph.1995.1039. URL http://linkinghub.elsevier.com/retrieve/doi/10.1006/jcph.1995.1039. [p.25], [p.106], [p.112]

[168] Steve Plimpton, Roy Pollock, and Mark J. Stevens. Particle Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations. In *Proceedings of the Eighth Siam Conference on Parallel Processing for Scientific Computing*, pages 1–13. Citeseer, 1997. doi: 10.1.1.41.5174. [p.112]

[169] Steve Plimpton, Paul Crozier, and Aidan Thompson. Lammps-large-scale atomic/molecular massively parallel simulator. *Sandia National Laboratories*, 18, 2007. [p.11], [p.106]

[170] EL Pollock and Jim Glosli. Comments on p3m, fmm, and the ewald method for large periodic coulombic systems. *Computer Physics Communications*, 95(2-3): 93–110, 1996. [p.27]

[171] Matej Praprotnik, Silvina Matysiak, Luigi Delle Site, Kurt Kremer, and Cecilia Clementi. Adaptive resolution simulation of liquid water. *Journal of Physics: Condensed Matter*, 19(29):292201, 2007. [p.21]

[172] Charles M Rader. Discrete fourier transforms when the number of data samples is prime. *Proceedings of the IEEE*, 56(6):1107–1108, 1968. [p.46]

[173] Anthony K Rappé, Carla J Casewit, KS Colwell, WA Goddard Iii, and WM Skiff. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American chemical society*, 114(25):10024–10035, 1992. [p.19]

[174] Stephane Redon, Nico Galoppo, and Ming C Lin. Adaptive dynamics of articulated bodies. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 936–945. ACM, 2005. [p.29]

[175] Romain Rossi, Mathieu Isorce, Sandy Morin, Julien Flocard, Karthik Arumugam, Serge Crouzy, Michel Vivaudou, and Stephane Redon. Adaptive torsion-angle quasi-statics: a general simulation method with applications to protein structure analysis and design. *Bioinformatics*, 23(13):i408–i417, 2007. [p.20], [p.29]

[176] Benoıt Roux and Thomas Simonson. Implicit solvent models. *Biophysical chemistry*, 78(1-2):1–20, 1999. [p.21]

[177] Payam Rowghanian and Alexander Y Grosberg. Force-driven polymer translocation through a nanopore: An old problem revisited. *The Journal of Physical Chemistry B*, 115(48):14127–14135, 2011. [p.128], [p.129]

[178] John W Ruge and Klaus Stüben. Algebraic multigrid. *Multigrid methods*, 3(13): 73–130, 1987. [p.49]

[179] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003. [p.50]

[180] C. Sagui and T. Darden. Multigrid methods for classical molecular dynamics simulations of biomolecules. *Journal of Chemical Physics*, 114(15):6578–6591, 2001. ISSN 00219606. doi: 10.1063/1.1352646. [p.96]

[181] Celeste Sagui and Thomas Darden. Multigrid methods for classical molecular dynamics simulations of biomolecules. *The Journal of Chemical Physics*, 114(15): 6578–6591, 2001. [p.27]

[182] Takuya Saito and Takahiro Sakaue. Cis-trans dynamical asymmetry in driven polymer translocation. *Physical Review E*, 88(4):042606, 2013. [p.129]

[183] Jesus M Sanz-Serna. Symplectic integrators for hamiltonian problems: an overview. *Acta numerica*, 1:243–286, 1992. [p.9]

[184] Jalal Sarabadani, Timo Ikonen, Harri Mökkönen, Tapio Ala-Nissila, Spencer Carson, and Meni Wanunu. Driven translocation of a semi-flexible polymer through a nanopore. *Scientific Reports*, 7(1):7423, 2017. [p.129]

[185] Tamar Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*, volume 21. Springer-Verlag New York, Inc., 2010. ISBN 978-1-4419-6350-5. doi: 10.1007/978-1-4419-6351-2. [p.96]

[186] Tamar Schlick. *Molecular modeling and simulation: an interdisciplinary guide: an interdisciplinary guide*, volume 21. Springer Science & Business Media, 2010. [p.14], [p.16], [p.17]

[187] Kevin E Schmidt and Michael A Lee. Implementing the fast multipole method in three dimensions. *Journal of Statistical Physics*, 63(5-6):1223–1235, 1991. [p.27]

[188] Hans Rudolf Schwarz and Jörg Waldvogel. *Numerical analysis: a comprehensive introduction*. John Wiley & Sons, 1989. [p.101]

[189] Yair Shapira. Note on the multigrid w-cycle. *Journal of Computational and Applied Mathematics*, 85(2):351 – 353, 1997. ISSN 0377-0427. doi: https://doi.org/10.1016/S0377-0427(97)00142-8. URL http://www.sciencedirect.com/science/article/pii/S0377042797001428. [p.52]

[190] Krishna Kant Singh and Stephane Redon. Adaptively Restrained Molecular Dynamics in LAMMPS. *Modelling and Simulation in Materials Science and Engineering*, 2017. ISSN 1361651X. doi: 10.1088/1361-651X/aa7345. [p.29], [p.31], [p.67], [p.103]

[191] Krishna Kant Singh and Stephane Redon. Single-pass incremental force updates for adaptively restrained molecular dynamics. *Journal of Computational Chemistry*, 2017. ISSN 1096-987X. doi: 10.1002/jcc.25126. URL http://dx.doi.org/10.1002/jcc.25126. [p.29], [p.64], [p.67], [p.105]

[192] Krishna Kant Singh, Dmitriy F Marin, and Stephane Redon. Parallel adaptively restrained molecular dynamics. In *High Performance Computing & Simulation (HPCS), 2017 International Conference on*, pages 308–314. IEEE, 2017. [p.29], [p.35], [p.108], [p.116]

[193] Kyaw Sint, Boyang Wang, and Petr Král. Selective ion passage through functionalized graphene nanopores. *Journal of the American Chemical Society*, 130(49):16448–16449, 2008. [p.110], [p.119]

[194] Robert D Skeel, Ismail Tezcan, and David J Hardy. Multiple grid methods for classical molecular dynamics. *Journal of Computational Chemistry*, 23(6):673–684, 2002. [p.27]

[195] Robert D Skeel, David J Hardy, and James C Phillips. Correcting mesh-based force calculations to conserve both energy and momentum in molecular dynamics simulations. *Journal of computational physics*, 225(1):1, 2007. [p.74], [p.112]

[196] Elena Slonkina and Anatoly B Kolomeisky. Polymer translocation through a long nanopore. *The Journal of chemical physics*, 118(15):7112–7118, 2003. [p.126]

[197] Volker Springel, Simon DM White, Adrian Jenkins, Carlos S Frenk, Naoki Yoshida, Liang Gao, Julio Navarro, Robert Thacker, Darren Croton, John Helly, et al. Simulations of the formation, evolution and clustering of galaxies and quasars. *nature*, 435(7042):629, 2005. [p.4], [p.5]

[198] Gabriel Stoltz and Zofia Trstanova. Stable and accurate schemes for langevin dynamics with general kinetic energies. *arXiv preprint arXiv:1609.02891*, 2016. [p.136]

[199] Arnold J Storm, JH Chen, HW Zandbergen, and C Dekker. Translocation of double-strand dna through a silicon oxide nanopore. *Physical review E*, 71(5):051903, 2005. [p.129]

[200] WB Streett, DJ Tildesley, and G Saville. Multiple time-step methods in molecular dynamics. *Molecular Physics*, 35(3):639–648, 1978. [p.26]

[201] Markus Stürmer, Harald Köstler, and Ulrich Rüde. A fast full multigrid solver for applications in image processing. *Numerical linear algebra with applications*, 15 (2-3):187–200, 2008. [p.49]

[202] Myung E Suk and NR Aluru. Water transport through ultrathin graphene. *The Journal of Physical Chemistry Letters*, 1(10):1590–1594, 2010. doi: 10.1021/jz100240r@proofing. URL http://dx.doi.org/10.1021/jz100240r@proofing. [p.110]

[203] Huai Sun. Compass: an ab initio force-field optimized for condensed-phase applications overview with details on alkane and benzene compounds. *The Journal of Physical Chemistry B*, 102(38):7338–7364, 1998. [p.13]

[204] W Sung and PJ Park. Polymer translocation through a pore in a membrane. *Physical review letters*, 77(4):783, 1996. [p.128]

[205] Godehard Sutmann and Bernhard Steffen. A particle–particle particle-multigrid method for long-range interactions in molecular simulations. *Computer physics communications*, 169(1-3):343–346, 2005. [p.27]

[206] Godehard Sutmann, Paul Gibbon, and Thomas Lippert. *Fast Methods for Long-Range Interactions in Complex Systems*. Forschungszentrum Jülich, 2011. [p.27], [p.28], [p.96], [p.99], [p.107]

[207] Agilent Technologies. Agilent technologies, 2018. URL http://www.agilent.com. [p.127]

[208] Oxford Nanopore Technologies. Oxford nanopore technologies, 2018. URL https://nanoporetech.com/. [p.127]

[209] Ulrich Trottenberg, Cornelius W Oosterlee, and Anton Schuller. *Multigrid*. Academic Press, Inc., 2000. ISBN 012701070X. [p.49], [p.51], [p.52], [p.53], [p.54], [p.55], [p.96], [p.100], [p.107]

[210] Zofia Trstanova. *Mathematical and algorithmic analysis of modified Langevin dynamics*. PhD thesis, Université Grenoble Alpes, 2016. [p.33], [p.136], [p.137]

[211] Zofia Trstanova and Stephane Redon. Estimating the speed-up of adaptively restrained langevin dynamics. *Journal of Computational Physics*, 336:412–428, 2017. [p.29], [p.31], [p.108]

[212] MBBJM Tuckerman, Bruce J Berne, and Glenn J Martyna. Reversible multiple time scale molecular dynamics. *The Journal of chemical physics*, 97(3):1990–2001, 1992. [p.9]

[213] Are Magnus Bruaset Aslak Tveito and Are Magnus Bruaset. *Numerical solution of partial differential equations on parallel computers*. Springer, 2006. [p.55]

[214] Patrick T Underhill and Patrick S Doyle. On the coarse-graining of polymers into bead-spring chains. *Journal of non-newtonian fluid mechanics*, 122(1-3):3–31, 2004. [p.130]

[215] Adri CT Van Duin, Siddharth Dasgupta, Francois Lorant, and William A Goddard. Reaxff: a reactive force field for hydrocarbons. *The Journal of Physical Chemistry A*, 105(41):9396–9409, 2001. [p.12]

[216] Wilfred F van Gunsteren and Herman JC Berendsen. Computer simulation of molecular dynamics: Methodology, applications, and perspectives in chemistry. *Angewandte Chemie International Edition*, 29(9):992–1023, 1990. [p.26]

[217] Bala Murali Venkatesan and Rashid Bashir. Nanopore sensors for nucleic acid analysis. *Nature nanotechnology*, 6(10):615, 2011. [p.126]

[218] Loup Verlet. Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical review*, 159(1):98, 1967. [p.25]

[219] Junmei Wang, Romain M Wolf, James W Caldwell, Peter A Kollman, and David A Case. Development and testing of a general amber force field. *Journal of computational chemistry*, 25(9):1157–1174, 2004. [p.11]

[220] Meni Wanunu, Jason Sutin, Ben McNally, Andrew Chow, and Amit Meller. Dna translocation governed by interactions with solid-state nanopores. *Biophysical journal*, 95(10):4716–4725, 2008. [p.129]

[221] Dongshan Wei, Wen Yang, Xigao Jin, and Qi Liao. Unforced translocation of a polymer chain through a nanopore: The solvent effect. *The Journal of chemical physics*, 126(20):05B610, 2007. [p.128]

[222] David B Wells, Maxim Belkin, Jeffrey Comer, and Aleksei Aksimentiev. Assessing graphene nanopores for sequencing dna. *Nano letters*, 12(8):4117–4123, 2012. [p.126]

[223] P Wesseling. Cell-centered multigrid for interface problems. *Journal of Computational Physics*, 79(1):85–91, 1988. [p.54]

[224] P Wesseling. Introduction to multigrid methods. Technical report, INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING HAMPTON VA, 1995. [p.49], [p.55]

[225] D Wolf, P Keblinski, SR Phillpot, and J Eggebrecht. Exact method for the simulation of coulombic systems by spherically truncated, pairwise r- 1 summation. *The Journal of chemical physics*, 110(17):8254–8282, 1999. [p.28]

[226] Lijiang Yang, Chun-hu Tan, Meng-Juei Hsieh, Junmei Wang, Yong Duan, Piotr Cieplak, James Caldwell, Peter A Kollman, and Ray Luo. New-generation amber united-atom force field. *The journal of physical chemistry B*, 110(26):13166–13176, 2006. [p.20]

[227] Qiu Ying-Hua, Li Kun, Chen Wei-Yu, Si Wei, Tan Qi-Yan, and Chen Yun-Fei. Ion and water transport in charge-modified graphene nanopores. *Chinese Physics B*, 24(10):108201, 2015. [p.110]

[228] Rio Yokota and Lorena A Barba. A tuned and scalable fast multipole method as a preeminent algorithm for exascale systems. *The International Journal of High Performance Computing Applications*, 26(4):337–346, 2012. [p.27]

[229] Bojan Zagrovic, Eric J Sorin, and Vijay Pande. β-hairpin folding simulations in atomistic detail using an implicit solvent model1. *Journal of molecular biology*, 313(1):151–169, 2001. [p.22]